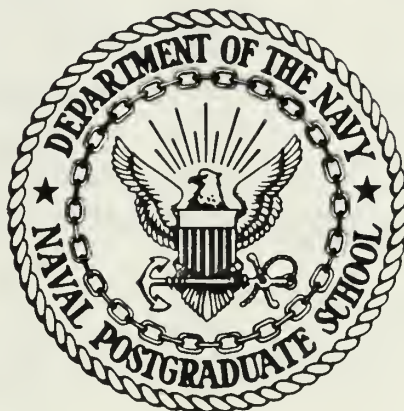


DO
NA
MONTEREY, CALIFORNIA 94043

NAVAL POSTGRADUATE SCHOOL

Monterey, California



THESIS

THE FEASIBILITY OF IMPLEMENTING AN EXPERT
SYSTEM FOR AIRCRAFT MAINTENANCE
DISCREPANCY SCHEDULING WITH THE
NAVAL AVIATION LOGISTICS
COMMAND MANAGEMENT
INFORMATION SYSTEM
(NALCOMIS)

by

Martin J. McCaffrey

September 1985

Thesis Co-Advisors:

N.R. Lyons

D.R. Dolk

Approved for public release; distribution is unlimited

T224181

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) The Feasibility of Implementing an Expert System for Aircraft Maintenance Discrepancy Scheduling with the Naval		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis September 1985
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Martin J. McCaffrey		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93943-5100		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93943-5100		12. REPORT DATE September 1985
		13. NUMBER OF PAGES 127
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution is unlimited		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Expert Systems; Knowledge-Based Systems; NALCOMIS; Aviation Maintenance Scheduling; Knowledge Acquisition; NALDA; Heuristic		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) The feasibility of developing an expert system to support the scheduling of discrepancies in Maintenance Control is examined. A general review of expert systems and NALCOMIS is presented. An in-depth analysis of the scheduling and planning process is made. This analysis is based on interviews with several experts. The ability of NALCOMIS to support an		

#4 - TITLE - CONTINUED

Aviation Logistics Command Management Information
System (NALCOMIS)

#19 - ABSTRACT - CONTINUED

expert system and discussion on whether such a system
is warranted for this problem domain conclude the
thesis.

Approved for public release; distribution is unlimited.

The Feasibility of Implementing an Expert System
for Aircraft Maintenance Discrepancy Scheduling
with the Naval Aviation Logistics Command
Management Information System (NALCOMIS)

by

Martin J. McCaffrey
Major, United States Marine Corps
B.A., California State University, Fullerton, 1978

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN INFORMATION SYSTEMS

from the

NAVAL POSTGRADUATE SCHOOL
September 1985

740015
M1634

ABSTRACT

The feasibility of developing an expert system to support the scheduling of discrepancies in Maintenance Control is examined. A general review of expert systems and NALCOMIS is presented. An in-depth analysis of the scheduling and planning process is made. This analysis is based on interviews with several experts. The ability of NALCOMIS to support an expert system and discussion on whether such a system is warranted for this problem domain conclude the thesis.

TABLE OF CONTENTS

I.	INTRODUCTION -----	10
II.	KNOWLEDGE-BASED SYSTEMS -----	13
	A. CATEGORIES OF KNOWLEDGE SYSTEMS -----	15
	B. STAGES OF DEVELOPMENT -----	17
	C. COMPONENTS -----	17
	1. Knowledge Base -----	20
	2. Inference Engine -----	24
	3. Natural Language -----	28
	D. EXPERT SYSTEM DEVELOPMENT LIFE CYCLE -----	29
	1. Problem Identification Stage -----	31
	2. Conceptualization Stage -----	35
	3. Implementing A Prototype -----	35
	4. Testing -----	37
	5. Revision and Expansion of the Prototype --	40
	6. Integration and Maintenance -----	42
	E. DEVELOPMENT LANGUAGES AND TOOLS -----	44
	1. Languages -----	44
	2. Development Tools -----	47
	F. SHORTCOMINGS -----	48
	G. CONCLUDING REMARKS -----	50
III.	MAINTENANCE CONTROL AND DISCREPANCY SCHEDULING ---	52
	A. AIRCRAFT READINESS REPORTING TERMS -----	54
	1. Mission Capable -----	54
	a. Full Mission Capable -----	54

b. Partial Mission Capable -----	55
2. Not Mission Capable -----	55
a. Not Mission Capable Maintenance -----	55
b. Not Mission Capable Supply -----	55
B. MAINTENANCE/MATERIAL CONTROL -----	56
C. DISCREPANCY SCHEDULING -----	61
1. Decision Environment -----	63
2. Priorities -----	66
a. Flight Schedule Commitments -----	66
b. Downing Discrepancies -----	67
c. Up Discrepancies -----	68
d. Scheduled Maintenance/High Time Components -----	70
e. Other Considerations -----	72
3. Constraints -----	74
a. Hangar Space -----	75
b. IMRL -----	75
c. PME/Test Equipment -----	75
d. GSE -----	76
e. Personnel -----	76
f. Type of Discrepancy -----	77
g. Local Constraints -----	77
h. Support -----	78
4. Knowledge Base -----	78
a. General Prioritization Heuristics ----	79
b. Aircraft Systems Knowledge -----	79

	c. Parts Availability -----	81
	d. Personnel Capabilities -----	82
5.	Conclusions/Comments -----	82
	a. Complexity of the Decision Process ---	82
	b. Difficulties Encountered -----	83
IV.	NALCOMIS -----	85
	A. HISTORY -----	86
	B. OBJECTIVES -----	87
	C. HARDWARE -----	89
	D. SOFTWARE -----	91
	1. Honeywell Software -----	91
	2. Application Software -----	92
	E. EXPANDABILITY -----	93
V.	A KNOWLEDGE BASE FOR THE MAINTENANCE SCHEDULING DOMAIN -----	94
	A. PLANNING/SCHEDULING DOMAINS -----	94
	B. KNOWLEDGE FROM THE PRESENT DATA BASE -----	95
	1. Aircraft Facts -----	96
	2. Other Facts -----	97
	C. KNOWLEDGE BASE -----	97
	1. Facts -----	97
	2. Heuristics -----	101
	D. KNOWLEDGE FROM USER QUERY -----	102
	E. KNOWLEDGE NOT CONSIDERED -----	104
	1. Personnel -----	104
	2. Common Sense Maintenance Knowledge -----	105

VI.	FEASIBILITY ANALYSIS -----	107
A.	EXPERT SYSTEM APPLICABILITY TO MAINTENANCE SCHEDULING -----	107
1.	Suitability of the Problem Domain -----	107
2.	Benefits from Developing an Expert System -----	109
3.	Drawbacks -----	111
B.	NALCOMIS CAPABILITY TO SUPPORT AN EXPERT SYSTEM -----	111
C.	IS DEVELOPMENT OF AN EXPERT SYSTEM JUSTIFIED? -----	114
VII.	CONCLUSIONS AND RECOMMENDATIONS -----	119
	LIST OF REFERENCES -----	123
	INITIAL DISTRIBUTION LIST -----	126

LIST OF FIGURES

2.1	Steps in Building an Expert System -----	18
2.2	Expert System Components -----	19
2.3	Knowledge Representation Schemes -----	22
2.4	Stages in an Expert System Development Life Cycle -----	45
3.1	O-Level Maintenance Department Organization -----	53
3.2	Major Responsibilities and Software -----	60
4.1	NALCOMIS Hardware and Software -----	90
5.1	Discrepancy Scheduling Rules -----	103

I. INTRODUCTION

The primary goal of Naval aviation maintenance is to provide the required mission capable aircraft to meet its squadron's commitments. To do this it seeks to maximize aircraft operational readiness. The Maintenance Control Officer, in prioritizing and assigning all work assignments, has a significant impact on this objective.

Maintenance control is one of the most demanding, complex, and mentally stressful work environments in the military today. It is the focal point where all support and information assets converge in a squadron.

It is the objective of the Maintenance Control Officer to optimize the use of available manpower and material resources in seeking maximum aircraft operational readiness. Optimum assignments and decisions require efficient processing of all available information. Major responsibility for meeting aircraft availability requirements to fill operational flight assignments rests here.

Decisions are made under extremely demanding and time sensitive conditions. Turnover of key personnel is relatively high compared to comparable civilian environments. Constraints restrict the decision making in this environment. Also, some question exists as to the ability of the decision makers to synthesize adequately all the information for effective decision making.

Development of expert or knowledge-based systems has rapidly expanded in the past few years. Such systems are knowledge-intensive programs which solve problems requiring human expertise. Developed applications include such areas as diagnosis, interpretation, and scheduling. It may be possible for such a system to provide decision support for job planning and scheduling in aviation maintenance control.

With the implementation of the Naval Air Logistics Command Management Information System (NALCOMIS), considerable computer resources will become available to every squadron in Naval aviation. The present configuration at the squadron level includes a Honeywell DPS 6/54 minicomputer with one megabyte of memory and three Winchester 52 megabyte disk drives. NALCOMIS provides a real-time management information system for aviation maintenance. No provision is currently made, however, to provide any enhanced decision support capability with the system.

Because of the possible benefits an expert system offers for improving the decision making effectiveness in this area, and the potentially improved operational readiness that would result, an investigation of the feasibility of applying this technology to the scheduling of work assignments is warranted. The purpose of this thesis is to evaluate the feasibility of employing an expert system for scheduling and prioritizing aircraft maintenance work assignments and to

consider the implementation of such a system using NALCOMIS assets.

Chapter II provides an overview of current expert system technology and practice. The elements, components, and theory behind such systems are discussed. Chapter III details the functions and structure of an organizational level maintenance department and provides information on the responsibilities and activities of the maintenance control officer and of maintenance control. It also provides an in-depth examination of the decision scheduling process. This information is the result of interviews with several experienced and "expert" maintenance control personnel. The knowledge requirements, constraints, and environmental factors are analyzed.

A comprehensive historical and technical discussion on NALCOMIS is included in Chapter IV. This material is oriented to the hardware and software currently implemented in the organizational level prototype system. Chapter V presents the knowledge base requirements for an expert system used in prioritizing aircraft discrepancy assignments. Available resources and needs are addressed. Chapter VI analyzes the feasibility of developing an expert system and NALCOMIS's ability to support such a system. Chapter VII summarizes the research and makes several recommendations based on the analysis.

II. KNOWLEDGE-BASED SYSTEMS

This chapter serves as an introduction to expert systems, or knowledge-based systems as they are sometimes called. Its purpose is to describe the concepts and basic elements of which a knowledge-based system is composed. General information, categories, stages in development, basic concepts, and components of these systems will be covered. Also discussed are knowledge representation, knowledge acquisition, and the benefits and shortcomings of these systems.

What is an expert or knowledge-based system? Weiss and Kulikowski [Ref. 1:p. 1] define an expert system as one that:

- . . . handles real-world, complex problems requiring an expert's interpretation.

- . . . Solves these problems using a computer model of expert human reasoning, reaching the same conclusions that the human expert would reach if faced with a comparable problem.

Another definition is offered by Professor Edward Feigenbaum [Ref. 2:p. 5], a leading researcher in the field of artificial intelligence (AI):

- . . . an intelligent computer program that uses knowledge and inference procedures to solve problems that are difficult enough to require significant human expertise for a solution.

Feigenbaum goes on to state that facts and heuristics make up the knowledge of an expert. "Facts" are public information generally agreed upon by experts in a field. "Heuristics" are rules of good judgment. Others often refer to

heuristics as "rules of thumb." Heuristics are not always widely known in a given domain and may vary from one expert to another.

For the past fifteen years, research scientists have been working intensely on projects in which they use domain specific knowledge as the basis for solving problems [Ref. 3: p. 264]. This is a different approach from earlier research which tried to program general problem solving strategies and failed [Ref. 4]. As a result, laboratory knowledge systems have demonstrated the ability to solve complex problems in scientific, medical, educational, business, and military applications.

In the past few years, some of these systems have found their way into the civilian market place. One of the first successful experimental systems was MYCIN. It diagnosed certain infectious blood diseases and recommended appropriate treatment. There are several commercial products based on MYCIN technology [Ref. 5]. R1, which is now known as XCON, is an expert system which configures Digital Equipment Corporation computer systems [Ref. 6]. Another system, Prospector, is used in geological evaluation of mineral deposits and has discovered deposits worth much more than its development costs [Ref. 7]. Stanford University, MIT, and Carnegie Mellon University have led research efforts in the area of expert systems research.

Barr and Feigenbaum have coined the term "knowledge engineering" to describe the field of AI involved with

solving problems that normally require human intelligence [Ref. 8]. The researchers who develop knowledge systems are called knowledge engineers.

A. CATEGORIES OF KNOWLEDGE SYSTEMS

Researchers have classified knowledge systems applications into several types or categories. Diagnostic systems deduce system malfunctions from observations. This category is the most prevalent of expert systems currently developed. Medical and electronic diagnostic systems have been fully developed and a majority of the commercially viable expert system tools have sprung from this type of system. MYCIN is the best known example of a diagnostic system.

Another type of system deals with prediction. Many of the same knowledge representations and control strategies that are used in diagnostic systems are applicable to this category. Prediction systems try to determine the consequences from a given set of facts and situation. Military and weather forecasting are two applications of this type of system.

Satisfying the constraints of a problem with a proper configuration falls under the design system category. Hayes-Roth states that such systems verify that a configuration, determined by a relationship of objects, meets the given constraints [Ref. 9]. Circuit design and budgeting are two areas that come under this category. Planning is considered a subgroup of the design problem. A planning system seeks

to deduce actions and their effects. Automatic programming, project, and military planning are examples of planning system applications.

Interpretation systems attempt to infer a situation description from observed facts. Intelligence analysis and chemical structure analysis are areas in which this type of system has been applied.

Several other categories have been established; however, no systems for these categories have gone past the laboratory development stage. Monitoring, debugging, instruction, repair, and control systems all fit this status. Extensive research work is being done in each of these categories. The potential benefits of an expert system monitoring the systems in a nuclear power plant or providing air traffic control have tremendous potential benefits.

Later discussion will cover knowledge representation and control schemes as well as development tools for knowledge systems. It should be mentioned that the type of problem to be solved should be matched with the most efficient techniques for developing that category of knowledge system.

This thesis is evaluating the feasibility of developing an expert system for scheduling aircraft maintenance discrepancies. This falls under the planning category mentioned earlier.

B. STAGES OF DEVELOPMENT

After many years of research, an iterative development process has emerged as the prevalent method for knowledge system construction. The following is a general summary of this iterative process. After determining that a problem can be reduced to a workable domain, an expert source of knowledge must be found. Initial interviews are conducted and a knowledge representation scheme is chosen. Reasoning strategies are then selected. The next step is to build a prototype. The prototype is then tested on cases developed by the expert. Modifications, and perhaps expansion, are made to the prototype, as necessary. This iterative process will be continued until the prototype produces what is considered expert status for the test cases. The prototype is then tested against actual field cases. Additional modifications can again be expected. Figure 2.1 depicts this development process.

C. COMPONENTS

Before one can hope to understand how an expert system is built, it is first necessary to have a knowledge of the major components that comprise such a system. Several components combine to form a generic knowledge system: the knowledge base, the inference engine, and the natural language user interface. Figure 2.2 is a diagram of the major components. Other elements to consider when building a knowledge system include the work area, control strategy,

DETERMINE PROBLEM DOMAIN/APPLICATION

FIND AN EXPERT

CONDUCT INITIAL INTERVIEWS

SELECT REASONING AND REPRESENTATION SCHEME

BUILD A PROTOTYPE

TEST CASE EVALUATION OF PROTOTYPE

MODIFY PROTOTYPE

FIELD TEST EVALUATION OF PROTOTYPE

MODIFY PROTOTYPE

TEST AND MODIFY AS NECESSARY

DELIVER SYSTEM

Figure 2.1 Steps in Building an Expert System

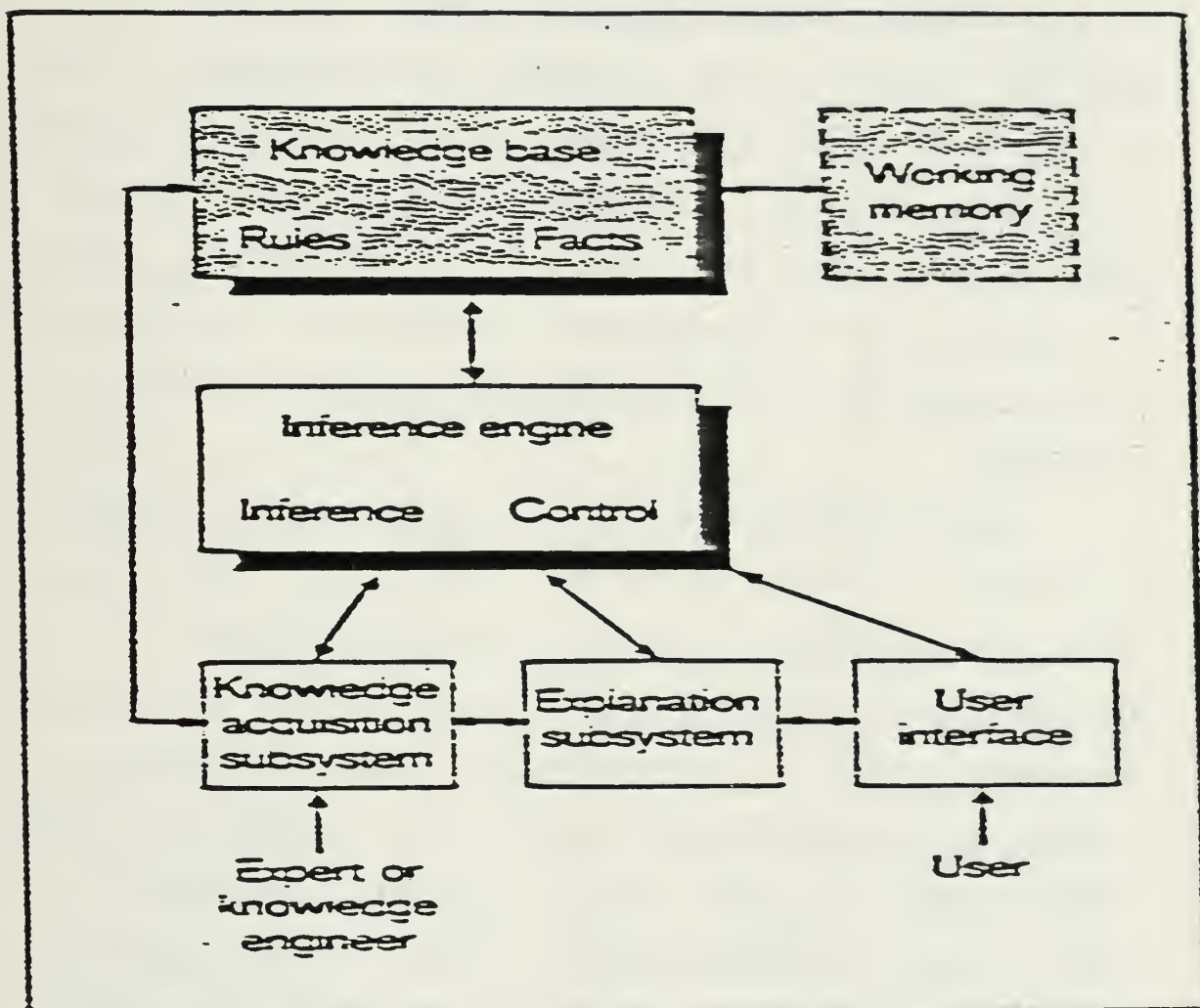


Figure 2.2 Expert System Components

knowledge acquisition, and explanation subsystems. Each of these elements will be covered in the paragraphs which follow.

1. Knowledge Base

This part of the system contains the experts' facts and heuristic rules that apply to the given problem domain. The way facts and relationships are encoded is known as knowledge representation. Knowledge representation is the major difference between knowledge-based systems and standard algorithmic programs. Knowledge representation is the key issue in how systems are built and how they perform. There are several methods of representing knowledge. In this section an overview of how this knowledge is represented is discussed.

The first representational scheme to be discussed is the semantic network. A semantic network uses nodes and links to represent abstract relations among objects in a knowledge base. An object, which may be either a physical or conceptual entity, is represented by a node. Elementary objects may be represented by alphanumeric characters called "atomic symbols." Nodes may also represent descriptors which provide information about objects. Links represent relationships between objects and descriptors. Two widely used links are isa and hasa, through which a graphic taxonomy is possible. Heuristic knowledge and definitional information are also provided by other links. The ability of nodes to inherit the characteristics of other related nodes in

their hierarchy, and the flexibility of this representational method, are its major advantages. Figure 2.3a is a simple semantic network representation.

A derivative of the semantic network is the object-attribute-value representation (O-A-V). As with semantic networks, objects are either physical or conceptual entities. Attributes are characteristics of objects, and value refers to the value of an attribute at a specific time (Figure 2.3.b). The only link relationships used are isa and hasa. These links join the object to the attribute and the attribute to the value, respectively. In the O-A-V representation scheme objects may be related, certainty factors may be used to indicate degrees of uncertainty. Trees are used to designate the order and relationship of objects [Ref. 2:p. 40].

A third method of representing knowledge is by using rules (Figure 2.3.c). This has been the dominant form of symbolic knowledge representation in first generation knowledge systems. A rule has a premise (the if clause) and a conclusion (the then clause). Logical connectives "AND" and "OR" are used to connect several clauses in a premise or a conclusion. Rules may vary from simple to complex. Uncertainty and variability may also be expressed.

Rules are widely used for representing procedural knowledge or methods of accomplishing goals. In solving problems, minimal coupling or interaction between rules



object	attribute	value
--------	-----------	-------

```

IF:      discrepancy is PMC, and
        it is in critical PMC list
THEN:    priority rating is medium

```

OBJECT: BOOK

ENTRIES

d. Frame Representation for the Concept of a Book

Figure 2.3 Knowledge Representation Schemes

exists. Using a rules scheme has the advantage of simplifying the generation of system prompts. A rule predicate is easily turned into a question. Likewise, explanation generation is also simplified.

Another way to represent knowledge in a knowledge base is by using frames [Ref. 10:pp. 73-77]. This representation is essentially a semantic network in which a frame is a description of an object that in turn provides slots for storing values associated with the object. Facts may be stated/stored by procedural or declarative representations. Slots may also include rules or default values about an object. Objects may inherit the attributes of more abstract objects. The primary advantage of frame representation is its ability to concisely store a great amount of knowledge about object properties and relations. Figure 2.3.d shows a frame representation.

The final representation method presently used in a knowledge base is logic-based. This method has many similarities to rule-based systems. First order predicate calculus is a formal way of representing logical propositions and the relationships between propositions. A predicate is a statement about an object or objects. A statement has a value of either true or false and can be linked into more complex expressions with connectives. Facts may be derived by applying basic rules of logic to the expressions. A primary advantage of logic representation is its ability to

represent almost any type of knowledge explicitly. Reference 11 provides a straightforward and detailed discussion on this representation.

Although early knowledge systems used only one of the previously discussed schemes for representing knowledge, more recent systems often use a combination of representations. Each method is used for the knowledge it represents best.

2. Inference Engine

This component embodies the control and inference strategies that experts apply when they manipulate the rules and facts stored in the knowledge base. It serves as the general reasoning mechanism and rule interpreter of the system. Two major jobs are performed by this component. If possible, it determines new facts by examining the facts and rules that exist at a given time. Secondly, it also decides the order of inferences. An inference is the process of deriving new facts from already known facts.

There are three inference strategies used in current rule and logic-based expert systems. The rule of logic commonly referred to as modus ponens is the most widely used strategy. It allows one to determine new facts from rules and known facts. In general this law states that if the premise of a rule is true, the conclusion may be assumed to be true, i.e., if A, then B.

A second inference strategy allows for the use of uncertain information in both facts and rules. Certainty factors are used by the inference engine for expressing indefinite or uncertain information.

Resolution is the third strategy and is a rather complex logical process. It basically condenses to the following: IF-THEN statements may be written as OR statements, and OR statements may be combined. Under appropriate circumstances, the inference engine uses this strategy in addition to modus ponens. See Reference 2, pp. 52-54 for more information.

In the present state of knowledge system development, these are the three basic inference strategies used. Although other strategies exist, they have not yet made it from the research laboratories to commercially available systems.

A knowledge system must have some way of determining where to begin the reasoning process, e.g., which rule to look at first. Secondly, there must be some way of resolving a situation where alternative lines of reasoning occur. Control mechanisms serve to satisfy these two requirements.

Two primary reasoning mechanisms are employed in present control techniques. These procedures are commonly referred to as backward chaining and forward chaining. The representation of the knowledge base is often the determining factor as to which of these implementations is used. When a

problem is being studied, the conceptual area defined by all the possible states that could occur as a result of interactions between the elements and operators is called the search space [Ref. 2:p. 55]. The shape of the search space, i.e., whether all the possible goals are known, frequently determines which method is more efficient. If the goal(s) state is not known forward chaining is used.

Backward inferencing occurs when the system works backward from a hypothetical solution or goal to determine evidence supporting the solution. Intermediate hypotheses are often formulated and tested during this process. A search of the knowledge base first centers on a rule or rules whose firing would give the desired conclusion [Ref. 12]. The premise, or antecedent part of the rule, will then focus on the problem facts stored in working memory. When this search finds a match with the antecedent rule the search is complete. If a search fails, the system will continue the search for another rule whose firing satisfies the first rule. This process continues until either a rule is found to satisfy the initial antecedent or the system asks for information from the user in an attempt to satisfy the rule. This type of inferencing is most effective when the possible outcomes are known and relatively small in numbers. It is the primary control strategy used in MYCIN [Ref. 13:p. 5].

Forward inferencing attempts to reason forward from the facts to a solution. This control technique is

data driven and is a more complex process than backward chaining.

A workarea is an area of memory set aside for storing a description of a problem constructed by the system from the facts supplied by the user or inferred from the knowledge base. This area is also called working memory. It contains the known facts. Forward chaining proceeds by first recognizing the rules that are satisfied based on the contents of working memory. The conclusions of such rules are then placed in the workarea. The system then checks additional rules, trying to determine which can succeed.

Other processes are also used by the inference engine for the purpose of control. Depth-first search of the knowledge base seeks to produce subgoals and pursues these goals until all information on them has been obtained. It seems to be the preferred method. Breadth-first search will first look at all the premises in a rule. This type of process may appear to be disjointed, especially if the user is asked for input. King and Harmon use an analogy that general problem solvers tend to use breadth-first reasoning because they inquire in a general way about the problem to be solved [Ref. 2:pp. 57-58]. On the other hand, a specialist problem solver concentrates on a specific aspect of the problem and looks for the details associated with one aspect at a time. A depth-first method is more appropriate in this case.

Monotonic and nonmonotonic reasoning are two other aspects often associated with the inference engine. With monotonic reasoning, all values for an attribute (a general characteristic or property of a physical entity) remain the same for the entire problem solving session. Thus the amount of true information continually grows. This type of reasoning is predominant in present systems.

Nonmonotonic reasoning allows facts that have initially true values to be changed. This is a very complex process to deal with in a computer environment. Planning is an example of one area where this type of reasoning may be used. Decisions which were originally believed to be true may be later retracted as additional information is determined.

This concludes discussion of inference engines and the control procedures associated with them. They are the state of the art and are being used in most of today's knowledge systems. More efficient methods of reasoning and search are still being sought, however, this is one reason why expert systems today are still restricted to specific problem domains and have not reached the capabilities of reproducing the broader knowledge and reasoning abilities of even a small child.

3. Natural Language

As this subject constitutes a major branch of AI research (as do expert systems), only a brief discussion dealing with the role of Natural Language in expert systems

will be made. Natural Language is basically a method that allows conventional language inputs to the computer system. Natural Language is usually introduced as a feature of a development tool. Its primary use in knowledge systems is to provide an interface for the user in developing knowledge bases. It must be stressed that these conventional language inputs are not to the stage of development that any semantic input is acceptable. Rather, a structured and somewhat constrained English language input and output is used. The use of Natural Language has proven quite beneficial in building knowledge bases using knowledge development tools.

D. EXPERT SYSTEM DEVELOPMENT LIFE CYCLE

The term "knowledge acquisition" refers to the process used by a knowledge engineer of extracting knowledge from an expert source and programming it in the knowledge system. Human experts are not the only sources of knowledge. Text-books, empirical data, databases, and other human non-experts are examples of other potential sources of knowledge. Knowledge acquisition is an extremely complex and somewhat ill-structured process that involves problem definition, implementation, and refinement, in addition to the representation of the expert's facts and relations in the knowledge base. Highly detailed and refined domain-specific knowledge is required to solve a difficult knowledge acquisition problem. One should remember that building expert systems is not

a well defined and understood development process. Only theoretical work has been done in some areas. However, the research completed over the last ten to fifteen years has provided a general sequence of stages that occur for most developments.

Locating, collecting, and refining knowledge are all part of the knowledge acquisition process. The gathered knowledge must be converted to an acceptable computer programming form. The knowledge acquisition process envelops all the stages to be discussed, i.e., it is not restricted to only one or two stages. Throughout the acquisition life cycle the knowledge engineer attempts to determine the procedures, specific facts, and judgmental rules of a domain that an expert uses to solve a problem.

Knowledge acquisition is the bottleneck in building expert systems [Ref. 14:p. 129]. Because knowledge acquisition has proven to be such an arduous and intricate procedure, few detailed accounts of the process have been written. The most cogent and specific coverage to be found was by Buchanan et al., in Hayes-Roth's Building Expert Systems [Ref. 14]. The reader should turn there for a more detailed account of the knowledge acquisition process.

As with other software development efforts, building an expert system lends itself to being an iterative process and knowledge engineers have divided it into several major stages. These stages need not proceed in the exact order

presented here. In fact, some of these stages will often be repeated several times during the development life cycle. The following is only a logical presentation of the stages.

1. Problem Identification Stage

One of the first steps in project development is to determine who will be the key players in the development of the expert system and what roles each will play. A project may involve one or more knowledge engineers. Although more than one expert can be chosen, this is not normally the case. The task of extracting the basic relations, concepts, and definitions related to the problem is quite difficult. The knowledge engineer must structure an expert's knowledge and help the expert to identify and formalize domain concepts [Ref. 14:p. 165]. This is a formidable task when only one expert is involved. Experience has shown that involvement of two or more experts intensifies the problems, since it is unlikely that two experts would be in total agreement on how to solve a problem. Trying to choose between differing views in the early development stages unduly complicates the problem and should be avoided. It has become standard practice to choose one expert to work with initially. Later in the development, other experts may be included to test and revise a newly developed system.

Knowledge engineers must first totally immerse themselves in a project, learning as much as possible about the problem and its domain. Ways of becoming more familiar

with a problem include studying readings and reports on the subject, talking to people knowledgeable in the field, or by visiting the site of the problem to be solved and observing, first hand, present procedures being used.

After this initial research, the knowledge engineer must determine if the problem is suitable for knowledge system application. This is one of the most critical points in the development cycle. Knowledge that is subjective, symbolic, changing or in part judgmental, is appropriate for expert system development. Knowledge that is stable, numerical, formalized, or firm is probably better implemented as a traditional algorithmic program.

If a problem proves suitable to knowledge system development, the knowledge engineer must begin considering the general type of task this problem falls under (i.e., monitoring, diagnosing, etc.), based on the initial information he has acquired. At this point, appropriate methods of representation and control must be considered.

During this stage an expert must be picked to work on the development project. In some situations there may be only one expert in the organization to consider. In others the search and selection process will be more difficult. By talking to others in the field, the knowledge engineer should be able to narrow the search process to a person whose performance and reputation in the problem domain clearly exceeds others--an expert.

Simply finding an expert candidate however, does not necessarily end the task. If a person is an expert in a given field, the firm may be reluctant to lose his expertise for the period of time necessary to fully develop an expert system--up to two years in some cases. If the system is to have a good chance of performing at a high level of expertise, however, the company must be convinced of the necessity of doing without this highly valued asset. Any less commitment on the part of management is certain to decrease the chances of producing a reliable system. Support for this issue is essential.

Another issue to be considered is the ability of the knowledge engineer and the expert to communicate with one another. These individuals will be working in a very complex environment for a lengthy period of time. Each must have confidence in the other's ability and be able to get along with the other. If this area is a troublesome one, consideration should be given to replacing either the knowledge engineer or the expert.

Having chosen the two key participants, the problem then needs to be fully defined. Problem characteristics and subproblems need to be determined. The terms, available data, and their relations must also be defined. Consideration should be given to what a solution to the problem contains. The present role of the expert in solving the problem should be evaluated. Also, the key concepts related

to the problem should be developed and clarified.

Buchanan classifies knowledge into two sets, strategic knowledge and structural knowledge [Ref. 14:p. 134]. Structural knowledge specifies the terms and tasks the user is determining; strategic knowledge provides how and when the system will establish these terms and tasks. He indicates these two types of knowledge are joined to what he terms the system's inference structure. This, or a similar organizational approach, is appropriate for development.

Another consideration during the problem identification stage is to determine the personnel, computing, and monetary requirements necessary to develop the knowledge system. The two primary personnel required for the project are the knowledge engineer and the problem expert. As mentioned previously, considerable time requirements for these players must be allowed. Development of a knowledge system may necessitate the dedication of considerable computer time and assets or even the acquisition of a separate computer environment. Procurement of a software development tool may also be required. All these costs need to be considered in this initial stage of development.

From the information determined in defining the problem, the goals or objectives of the system must be identified. Also any constraints to the project should be agreed on.

From the material presented thus far, it is easy to see the importance and complexity of this first stage in

developing a project. Hard, cold facts and issues need to be completely evaluated during this stage if an efficient development process is to be ensured.

2. Conceptualization Stage

The conceptualization stage consists of a detailed determination of the concepts of the problem and their relationships. Relationships between objects are stated. The processes involved in the problem solving and any constraints are settled. During this stage an attempt is made to separate and identify the knowledge needed for solving a problem from the knowledge used to justify a solution. At some point the concepts and relations should be written down and formalized. At this stage in the development cycle the knowledge engineer and the expert continue to have a close working relationship. The knowledge engineer will also continue thinking about which architectures are best suited to organizing the gathered knowledge, as well as appropriate tools that may be useful for representation. A primary goal of this stage is to reach the point where work on an initial prototype system can begin.

3. Implementing A Prototype

This stage consists of much more than prototype construction. Initially, the conceptual information is taken and put in a representative form that can be used with a chosen implementation tool. The initial prototype should not

attempt to encompass the entire problem domain, but rather only a limited but representative problem segment.

Selecting which tool to use will be greatly influenced by the inference strategies and knowledge employed by the expert in the process of solving a problem. Seldom is one tool advantageous in all areas. The tool with minimum disadvantages and a representational framework most applicable to the major areas of the problem is chosen. Harmon and King advocate that a primary conclusion of the prototyping be the adequacy of the selected expert system building tool for expressing the expert's knowledge and heuristics [Ref. 2:pp. 202-203].

In formalizing the problem, constructing a model of the problem solving process can be an important factor in development of a prototype system. Model types can be either mathematical or behavioral [Ref. 14:p. 145]. The role and characterization of data should also be carefully analyzed. Included are such considerations as the uncertainty, reliability, and consistency of the data. Partial specification of such information has proven very beneficial in previous development processes.

During this phase the knowledge engineer works closely with the expert, not only to extract the essence of the problem solving method and heuristics, but also to teach the expert how to formulate his views in rule forms. The knowledge engineer may possibly demonstrate how he converts the

expert's reasoning processes into rules of thumb to be used in the system. The expert will also be asked to formulate several test cases which may be used to test a broad range of system requirements. Typical elements to test include inference rules, control strategies and input/output outcomes. For instance, inference rules must be evaluated for correctness, consistency, and completeness. Test cases should be designed to test a broad range of requirements. Just as the prototype addresses only a subproblem of the domain, so a test case may be designed to test only specific aspects of a system.

The prototype system should include the data structures, inference strategies, and control techniques in a representational form expected to be used for total system development. Nevertheless, most authorities make a point of emphasizing that the initial prototype program should be designed from the standpoint that the entire program, or most of it, will be discarded and not used in the final product. The primary purpose of prototyping is to test the basic concepts, formalisms, and inference strategies the knowledge engineer has thus far developed, as well as test the design tool being used.

4. Testing

This step in the development cycle seeks to evaluate the accuracy and utility of the knowledge-based system. Testing should provide developers with the limitations of

the expertise of a system. A major goal of testing is to improve the design and performance of the expert system. Just as expert system technology is still developing, so are the methods of evaluating these systems. Testing as a methodology is rather primitive at this time. Nevertheless, many lessons in system evaluation have been learned and the fact that a state of the art has not yet been reached should not lessen the effort devoted to planned testing throughout the development stages.

Testing should be an integral part of the development process. Ideally, system evaluations should be first formulated when the system is being designed and should be conducted throughout the various stages of implementation. Unfortunately, this is frequently not the case. Gaschnig contends that planning tests early in development forces designers to determine specific system goals and objective measures for the achievement of those goals [Ref. 15:p. 243].

The formality and complexity of tests increases as system development progresses. The first testing is normally made after the initial prototype has successfully run the first couple of test cases and is initially an informal process. It concludes with formal structured evaluations of performance and user acceptability of the complete expert system.

Testing attempts to evaluate the functionality and accuracy of the knowledge base and inference structure. It

is much more than simply running test cases and comparing the results to those of the expert. For example, not only the fact that the right answer was derived, but the reason the system produced the right answer will be looked at. Typical characteristics of the knowledge system that are evaluated include the reliability of decisions and advice, correct reasoning, user requirements, ease of use, and input/output parameters. Program efficiency and the hardware environment should also be evaluated by testing.

When designing a test, a builder should keep in mind three key considerations: who it is for; what is to be evaluated; and the goals of the testing [Ref. 15:pp. 251-258]. An attempt to involve the user in the testing at an appropriate stage may pay considerable dividends to developers by giving early feedback on the user's likes or dislikes. User interface is normally not a primary concern during development of the initial prototype. Another benefit comes from the generation of user interest in a system and a feeling that user opinions are important to system development. This may lead to easier acceptance for the system when it is eventually introduced into the work environment.

Test cases that have successfully proven the capability of the system at one stage of development and that have themselves proven to be valid tests, should be repeated in each later test stage. This ensures that any additions or modifications to fix a problem have not caused new problems in areas formerly functioning properly.

Although the difficulty of designing effective evaluation criteria is apparent, this fact should not deter development of testing planning at an early stage. Testing plays a crucial role in the determination of the ultimate success of any knowledge system--its acceptance and use by the user.

5. Revision and Expansion of the Prototype

Although revision and expansion are listed together, they are, in fact, separate activities. Based on results of testing, the prototype and its successors are revised to meet the predetermined requirements. Facts and rules are revised as necessary. The development tool also is evaluated for its ability to provide the proper development environment during prototype implementation and testing. If the tool proves unsatisfactory, a new tool is selected and a new prototype built and tested.

Once the prototype is accurately functioning and has demonstrated the applicability of an expert system to the problem domain, work is begun on revising the prototype and developing a complete system. From the prototype, much insight is gained on the problem solving process and ways of representing the related knowledge and facts.

Because of basic design revisions, changes in facts, rules, and different hierarchial relationships, it is not unusual to discard the prototype and build the complete system from scratch using the lessons learned in the

prototype development. Recycling through the implementation and testing stages is common as the system is refined.

Development of the full expert system provides an expansion of the knowledge base in both depth and breadth. Considerable expansion and refinement of the heuristic rules are required. Not only are new rules added for covering other problems not originally represented by the prototype, but a finer, more in-depth knowledge in subproblem areas is included in the expanded system.

It is at this stage in the development cycle that intense effort is devoted to designing an expert system that interacts well with the user. A unique feature of expert systems is the explanation facility which is able to explain why it is seeking information or the basis of how it arrived at a decision. For these features to be useful to the user they must be easily accessible and concisely explain an action in English. This requires extensive effort on the part of the knowledge engineer and the expert during the design and programming of the system.

Revision, reimplementation, and testing continue until the knowledge engineer and expert agree the system is performing at an expert level. One final consideration is a decision on whether to use the system as developed in the unique knowledge engineering-based language or to convert the system to a more common application language for portability and integration with current hardware or databases.

At this point integration into the work environment is the next step.

6. Integration and Maintenance

This stage is another that is no less important than any of the others previously covered. No matter how good a knowledge system may be in giving correct answers and providing good advice, if it fails to gain acceptance by the user, it fails.

All the problems normally associated with introducing any new system into the workplace can be expected. The politics of orchestrating a major organizational change are bound to arise and must be dealt with. The knowledge engineer must attempt to foretell and take action to minimize such conflicts.

To overcome resistance to change, several things may be done. Prior planning that involves dissemination of information on the forthcoming system, opportunities for communications for those to be involved with the new system (both before and after introduction), and proper support for the new system are but a few. Extensive training for all involved with the system is also necessary if the maximum benefits are to be gained from the knowledge system and users are to be comfortable with its operation.

It is not unusual for any product involved with AI to initially meet with some degree of user resistance and skepticism. There may be several reasons for this and a

concerted effort must be made from the system's introduction to overcome this resistance. One approach is to emphasize that the expert system is being introduced not in an attempt to replace the decision making of the user, but rather as an aid to the user that may save time or replace burdensome tasks. Convincing a nonexpert to accept the expert system is essential.

Another consideration associated with the integration of the system is interfacing already existing systems or databases. Although planning for this should originate in earlier development stages, the actual interfacing takes place during this stage and may prove challenging [Ref. 16]. Even so, the fact that an organization already has data gathered in some organized manner should be considered as a source of facts for the expert system. Data should be viewed as a resource, and maximum use made of it.

Maintenance of a knowledge system varies from system to system. But like any software product, it is required and has considerable costs associated with it over the system's life. An expert system may be translated into a common language, such as BASIC or C, for improved efficiency or portability reasons. In such cases the local user has very limited maintenance capabilities. Any rule changes or additions are performed by the developers. In some cases, where the program is not translated into another language, the users may be allowed to make specified modifications,

which may include adding or modifying some rules. This provides the benefit of having an independent product, but requires more extensive training of users.

A major advantage of expert systems over algorithmic programs is modularity. For knowledgeable users of existing systems this has proven very beneficial and has reduced the complexity of changes since only a segment need be changed. Also, such modular changes do not affect other areas of the program. The stages in the development of an expert system are summarized in Figure 2.4.

E. DEVELOPMENT LANGUAGES AND TOOLS

This segment discusses the programming languages and tools used in the development of knowledge systems. LISP and PROLOG are the two symbolic programming languages most frequently used in AI. They have features which make it easier to build knowledge systems than do conventional languages which are designed for numerical operations. These two AI languages are more flexible than development tools, but also more difficult for prototyping a new system. In the past few years, several expert system building tools have become available. Such tools have incorporated basic knowledge engineering principles.

1. Languages

LISP is the language most frequently used for building knowledge systems. Essentially, LISP does not differentiate between data and programs. Only a few basic functions

PROBLEM IDENTIFICATION STAGE

CONCEPTUALIZATION STAGE

PROTOTYPE IMPLEMENTATION STAGE

TESTING STAGE

REVISION STAGE

EXPANSION STAGE

INTEGRATION STAGE

MAINTENANCE STAGE

Figure 2.4 Stages in an Expert System Development Life Cycle

are used. Storage space is efficiently managed and program modularity is a main feature. LISP's primary advantage over other high level languages, such as FORTRAN or PASCAL, is its ability to do symbolic processing. Symbolic programming provides manipulation of strings of symbols with logical, rather than numerical, operators.

Its greatest disadvantage is the requirement for a LISP interpreter. LISP interpreters are currently available for only a few computers. An interpreter serves to interpret the LISP functions so they may be executed by the hardware. Thus, an expert system written in a particular LISP language can only run on computers for which there is an interpreter for that language [Ref. 16]. As a result, a company may have to invest in a new computer in order to develop or implement a LISP-based expert system. LISP also suffers from a lack of standardization; several dialects exist.

Managers are frequently faced with the dilemma of distributing a runtime version in LISP or translating the LISP program into a more common language. The latter choice has the disadvantage of requiring an extensive time for re-write; however, this alternative may be cost effective if many users require the software and don't have LISP compatible hardware. This alternative also has the misfortune of requiring all maintenance to the program to be done by the developer.

Another representational language for encoding expert knowledge is PROLOG. This logic programming language was originally developed in Europe and is quite popular internationally. It is based on a subset of first order logic. Compared to conventional programming languages, its syntax requirements are much less complex. PROLOG has several distinctive features. These include that it is rule-based, it uses pattern matching, and it uses automatic backtracking [Ref. 17]. The same disadvantages associated with LISP are applicable to PROLOG.

2. Development Tools

A significant benefit of research over the past ten years has been the creation of expert system development tools that provide meaningful assistance in building expert systems. These tools are basically the frame or skeleton of an already existing expert system. The knowledge base, which contains the rules and data unique to a particular problem, is stripped away.

Expert system building tools do have several limitations. Present designs of such tools have only been able to capture certain types of knowledge that experts use in solving specific types of problems, such as diagnosis or prescription. Therefore one must insure a tool is appropriate to the problem prior to selecting a tool.

Some of these software tools are written in LISP or PROLOG and require AI capable hardware. Others have been

rewritten in conventional high level application languages. Each tool provides different methods of representing knowledge and inference. They are marketed by both universities and commercial companies, with the former's documentation generally less developed than the commercial product's. Support and training also tend to be better and more extensive for the commercial product.

Even with the noted disadvantages, expert system development tools have tremendous potential. King and Harmon assert that the majority of expert systems developed in the future will use expert system tools [Ref. 2:pp. 195-209]. In fact, they go so far as to state that if one is developing a large knowledge-based system and an expert system development tool is not available, most knowledge engineers would likely recommend discontinuation of the project. The reasons are the substantial cost and time required to develop an entire system from scratch.

F. SHORTCOMINGS

Although expert systems offer many benefits and have vast potential, there are several shortcomings which should be addressed. Development of such systems is not only difficult, but expensive and time consuming. Development and production costs are much higher than for other types of programming. Costs for existing systems have ranged from \$100,000 to over a million dollars.

Building knowledge systems is a lengthy process, especially if built from scratch without the aid of a development tool. Expert knowledge is often not well formulated and easily extractable. Initial systems required an average of twenty man years to develop, with more recent systems still requiring as many as five man years. Nevertheless, research and development of expert building tools can be expected to reduce this period in the future.

Although expert systems are solving problems that algorithmic programs could not, they do not have the capability of a human expert. These programs are taken from the deep knowledge of an expert; they consist of compiled surface knowledge. Explanations for their reasoning is rather shallow and novel situations are not solvable. Their ability to interact with the user is primitive compared to that of a human expert.

Presently, the most serious shortcoming in this field is the severe shortage of trained knowledge engineers able to develop these systems. Estimates have placed the number of knowledge engineers in the United States from 250 to 350. Most are working in academia, think tanks, or a few industrial labs. There are presently only a dozen or so commercial companies developing and marketing knowledge systems. Although the demand for knowledge base systems will continue to grow, the lack of knowledge engineers is a considerable constraint. Universities are not training many

of these engineers. In reality, the learning process for knowledge engineers has primarily been acquired by first-hand interviewing of experts. This is a slow process. It would seem that this shortage is likely to continue for the near future.

G. CONCLUDING REMARKS

This chapter has covered many aspects of expert system development. It was not intended to give detailed information on successfully developed systems and their technological approaches. This information is readily available from other sources. Rather, the intent was to convey information on the major components of knowledge-based systems and the approaches that make up knowledge system development.

No single development taxonomy has yet emerged to dominate this area. Many domain specific problem types have yet to submit to some symbolic programming solution. Although research is ongoing in these areas, progress is slow.

It should be stressed that although several systems are quite successful, just as in the case of most experts, these programs are not infallible. They do make mistakes. They also operate on complex problems at levels of success that equal or exceed the human expert they are designed to emulate. Chandrasedaran points out that the 80 percent/20 percent rule is quite applicable, i.e., it may be quite

reasonable to efficiently and economically capture 80 percent of the knowledge of a problem domain, but the remaining 20 percent may require trade offs which are unacceptable [Ref. 18]. These trade offs include such items as extremely high costs for the knowledge captured, extraordinary time requirements, or specifications which simply exceed the technological capabilities of the field.

Expert systems may have a bright future, but there are currently a number of constraints which restrict the growth of this technology. Hayes-Roth cites the shortage of skilled knowledge engineers, primitive development tools, and the difficulty of the work as reasons current demand of this technology exceeds supply [Ref. 19].

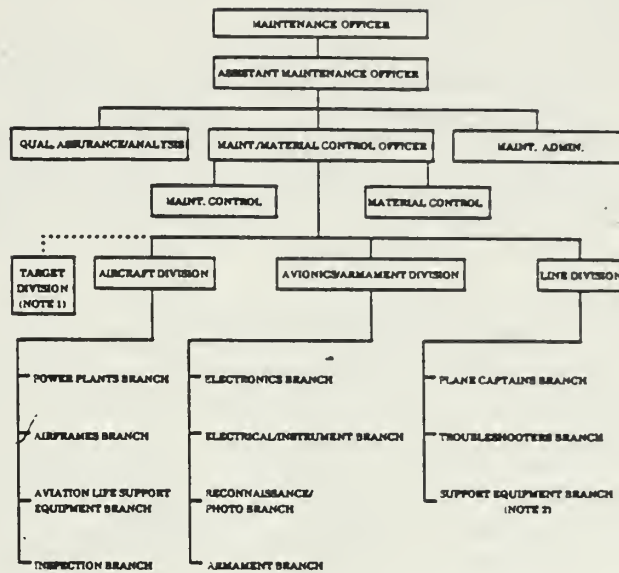
Knowledge systems will have a large impact on our society in the future. They offer tremendous promise for significant productivity increases in business. Some feel that development tools will become as common as many popular application programs such as Lotus 1-2-3 or VisiCalc are today ([Ref. 2:p. 253]). There is a vast potential for expert systems to revolutionize the use and benefits of computers to our society. The chapters which follow examine the feasibility of applying this technology to the aircraft maintenance discrepancy scheduling domain.

III. MAINTENANCE CONTROL AND DISCREPANCY SCHEDULING

The previous chapter presented the basic concepts and principles associated with expert systems and their development. This chapter examines Maintenance Control, the area of aircraft maintenance this study is evaluating for possible implementation of a knowledge-based system.

Organizational maintenance functions are assigned to squadrons by Reference 20, entitled the Naval Aviation Maintenance Program (NAMP). This is commonly referred to as O-level maintenance and basically consists of inspecting, troubleshooting, servicing and lubricating aircraft or aircraft systems. It also allows for the removal and replacement of parts and minor assemblies of the aircraft. Defective components are repaired at a higher level of maintenance.

To ensure effective management, the NAMP has assigned a standard organization for the O-level maintenance department. Figure 3.1 shows the organization for Navy and Marine Corps O-level units [Ref. 20:pp. 3-2-3]. It can be seen from this figure that these organizations differ only slightly. The organization is based on staff and line relationships. Line relationships are direct supervisory relationships; staff relationships are advisory in nature. Quality Assurance and Maintenance Administration are the staff divisions at the organizational level. The other work centers depicted



a. Navy O-Level Maintenance Organization



b. Marine Corps O-Level Organization

Figure 3.1 O-Level Maintenance Department Organization

have a line relationship. The central role of Maintenance Control is depicted in these charts.

Maintenance departments vary in size from 100 to 250 personnel. The number of personnel assigned depends on the number of aircraft assigned to a unit and the complexity of the aircraft.

The remainder of this chapter describes the responsibilities of the Maintenance Control Officer (MCO), followed by an examination of the prioritization of discrepancies. First, some terminology related to aircraft readiness status must be introduced.

A. AIRCRAFT READINESS REPORTING TERMS

Several terms are used specifically for describing the material condition of aircraft. These terms are defined in the following subparagraphs and are paraphrased from definitions specified in Reference 20, pp. C-32-33.

1. Mission Capable

The material condition of an aircraft which indicates it is capable of performing at least one and possibly more of its designated missions. A common term used to signify this condition is that the aircraft is "up," i.e., flyable. Mission Capable aircraft are divided into the following two categories.

a. Full Mission Capable (FMC)

The material condition indicating that an aircraft can perform all of its assigned missions.

b. Partial Mission Capable (PMC)

The material condition of an aircraft indicating that it can perform at least one, but not all, of its missions. This category is further broken into two subcategories. Partial Mission Capable Maintenance (PMCM) indicates that the reason for the PMC status is because of outstanding maintenance requirements which exist on the inoperable systems. The second subcategory is Partial Mission Capable Supply (PMCS), indicating that the PMC condition exists because maintenance cannot be performed because of a supply shortage of the required material.

2. Not Mission Capable (NMC)

The material condition of an aircraft which indicates it is unable to perform any of its missions. Aircraft in this category are commonly referred to as being "down," i.e., nonflyable. There are two subcategories for this status.

a. Not Mission Capable Maintenance (NMCM)

Indicates that the aircraft is down because of maintenance requirements.

b. Not Mission Capable Supply (NMCS)

The material condition of an aircraft which indicates it is not capable of performing any of its missions because the maintenance necessary to repair the discrepancy cannot continue because of a supply shortage of required material. Most maintenance personnel seldom

use these terms in their everyday discussions about aircraft. To a large extent they use the simple terminology "up" and "down" aircraft. An up aircraft is either FSC or PMC. A down aircraft is one which is not flyable and could more formally be described as either NMCM or NMCS.

B. MAINTENANCE/MATERIAL CONTROL

Maintenance Control is responsible for directing all aircraft maintenance activities within a squadron. It is the brain of the maintenance department, for from this work center all work is assigned and coordinated.

The Maintenance Control Officer heads this workcenter. Personnel staffing varies, depending on the number of aircraft assigned to a squadron and the manpower assigned to the maintenance department. Several maintenance controllers and an E-7 or E-8 Maintenance Control Chief (MCC) are usual staffing.

The NAMP sets forth many responsibilities for the MCO and Maintenance Control division. Among the primary responsibilities assigned the MCO are the control of the daily work load and assignment of work priorities for the maintenance department. This work center directs, coordinates, and monitors all maintenance actions, ensuring all resources of the department are used. Throughout the day the decisions that confront the MCO are complex and dynamic.

Assigning the necessary aircraft assets to meet the squadron's operational commitments is the overriding priority

each day. A typical flight schedule will involve aircraft launches in the morning, afternoon, and evening. Properly configured aircraft to complete the scheduled mission must be assigned. Frequently, not enough "up" aircraft are available to assign one to each scheduled mission. Work must then be directed to either turn around "up" aircraft that return from earlier missions or repair aircraft not initially capable of performing a given mission.

In order to maximize the aircraft operational readiness the MCO/MCC seek to effectively and efficiently manage the material and manpower resources available and direct these resources in a manner that yields the most "up" and full mission capable aircraft.

A key to the success of the maintenance department's effort to maximize operational readiness is the scheduling and prioritization of discrepancies to be worked on. Many factors enter into this decision process, including the available manpower and their qualifications, the expected required repair time, the availability of needed equipment and facilities, future commitments and deployments, etc. The pertinent factors involved in the decision process will be covered in detail below.

The MCO has considerable responsibilities in addition to those previously cited. These include the planning of the material support requirements for the department. Cannibalization control procedures, technical directive

incorporation, and scheduled maintenance planning are three other responsibilities.

Cannibalization is the process of removing a good part off one aircraft and the installation of this part on another aircraft for which the part is defective and not immediately available for issue from the supply system. Policies governing cannibalization must be followed and the tradeoffs carefully weighed. Scheduled maintenance is the periodic prescribed inspection or servicing of aircraft, normally done on a calendar of flight hour basis. This maintenance can be planned for in advance. It includes such maintenance as calendar and phased inspections or high time component removal.

Maintaining aircraft and equipment log books and weight and balance records are also the responsibility of the MCO. It also falls to him to maintain historical aircraft files and monitor 3M documentation.

VIDS boards and material requisitions must be validated daily by maintenance control. The responsibility for the establishment and maintenance of a tool control program is also assigned to the MCO, as well as formulation of the monthly maintenance plan.

As the central control point for maintenance, this center must constantly monitor and maintain cognizance of all uncompleted maintenance actions. This environment is constantly changing. New information is incessantly forthcoming. Additional new unscheduled discrepancies are the result of

returning aircraft missions where systems malfunctioned or from mechanics and technicians that discover faults during the normal course of their work.

The MCO is usually tasked with the responsibilities of operating the squadron material control center. This is the contact point within a maintenance department where material and parts requirements are coordinated with the local supply unit. Material control organizes the ordering, receipt, and delivery of parts and material. All NMCS and PMCS requisitions must be reconciled daily. Parts and material received from supply must be reconciled daily. Parts and material received from supply must be expeditiously distributed to the appropriate work centers.

In addition to the previous functions performed by maintenance control and material control, many unscheduled and unforeseen nonmaintenance-related requests center here. Personnel for work details, requests for tools or support equipment, and any number of additional inquiries are actions which must typically be handled during the course of the day.

Although this may seem to be just a long laundry list of requirements, each plays a necessary and important role to the overall management and operational success of the maintenance department.

Figure 3.2 summarizes the major responsibilities of the MCO and shows how diverse and demanding they are. It also

- * Assignment of Work Priorities
- * Control of Daily Work Load
- * Assign Aircraft to Meet Operational Flight Commitments
- * Effectively Manage Material and Manpower Resources
- * Control Cannibalization
- * Direct Scheduled Maintenance
- * Maintains Aircraft and Equipment Logs and Records
- * Establishes and Maintains the Tool Control Program
- * Responsible for Management of Material Control

Figure 3.2 Major Responsibilities of the MCO

serves to place the planning and scheduling of maintenance discrepancies into perspective with the many functions required of the MCO. It can be seen that planning is just one of many areas that requires action daily. Extensive amounts of time for this planning are simply not available when one considers all other requirements with which the MCO is tasked. A detailed examination of the decision scheduling process is the subject of the next section.

C. DISCREPANCY SCHEDULING

The order in which aircraft deficiencies are worked on, and the degree of utilization of the personnel, material, and equipment resources, have a key impact on a unit's aircraft operational readiness. If optimal scheduling decisions are made, the personnel, material, and equipment resources of the maintenance department are efficiently and effectively used to achieve maximum aircraft readiness. Poor scheduling decisions result in a decrease in combat readiness.

The prioritization of the discrepancies to be worked on is considerably more complex than it might seem at first. In this section the scheduling of discrepancies is carefully examined. General aspects of the process are covered first. The knowledge base that the decision maker uses is then examined, followed by the many factors and heuristics that enter into prioritization of discrepancies. The constraints that influence and restrict decision making are discussed

next, and closing comments are made on the process as a whole and the difficulty of acquiring the knowledge used in the decision process.

The material in this section was largely gathered through interviews with several professional maintenance personnel. They were selected on the basis of their experience in maintenance control and excellent professional reputations. They had recently served, or were serving, in the billet of MCO or MCC. On average they had ten to fifteen years experience in aircraft maintenance. Both senior enlisted and officers were interviewed.

The following list provides the objectives sought from the interviews.

- Observe the environment in which the scheduling decision is made.
- Determine the rules used by the MCO/MCC in scheduling priorities.
- Determine the knowledge base used by the MCO/MCC.
- Determine the constraints affecting the decision process.
- Determine how planning decisions were made.

The interview process was not unlike that experienced by the novice knowledge engineer whose investigation into the problem domain reaches the stage for the first interviews with the experts. This is categorized as part of the problem identification stage specified in Chapter II. In this case, an attempt to capture the fundamental considerations

and heuristics used in prioritizing maintenance discrepancy scheduling were the goals.

1. Decision Environment

The work priorities for the work centers are assigned at maintenance control meetings that occur two or three times a day. These meetings are headed by the MCO and the MCC; all work center supervisors attend.

The meetings include not only the scheduling of aircraft discrepancies, but also other information or tasks which may be pertinent to a work center. For instance, an aircraft may have to be towed to a particular area or hangar, or a work center tasked to configure an aircraft for a later mission, perhaps adding auxilliary fuel tanks. Information that is not related to maintenance on an aircraft, but which may affect the work center or department during the work day, is also covered. Examples are announcements of a squadron formation or air station quiet hours. Generally, flight schedule commitments, priority of discrepancies to be worked on, and any actions or activities that might affect a work center, its personnel, or the department are presented at these meetings.

Determination of the work priorities is normally made jointly by the MCO and the MCC. The MCC draws up the work schedule and then discusses it with the MCO. The MCC focuses primarily on meeting the immediate flight schedule commitments and maximizing the number of up aircraft. Knowledge and heuristics are used in arriving at a conclusion.

The MCO brings a longer range perspective to the process. In addition to the more immediate concerns of the MCC, considerations such as upcoming deployments, high time component changes, future mission requirements, and other department considerations often guide the MCO's decision making.

Most squadrons determine the general work plan two or three times during the course of the day. After the aircraft assignments are made to cover the morning flights, a general work priority schedule for the day is determined. This schedule lists the projected discrepancies to be worked on during the day. Emphasis is on the jobs scheduled for the morning. Discrepancies are assigned by work center and aircraft. Sufficient discrepancies to keep each of the work centers working through the morning are assigned.

This planning process normally requires from thirty to sixty minutes to draw up. Anywhere from twenty to forty discrepancies are assigned priorities and issued to work centers by order of precedence. These figures vary depending on the size and type of squadron and experience of the schedulers.

In many squadrons a modified work schedule is drawn up and a meeting held in the late morning. This modification incorporates new priorities resulting from new gripes from morning flights that have returned, as well as additional available information from discrepancy troubleshooting

performed in the morning. The late morning schedule primarily covers the work to be performed in the afternoon.

Another work schedule is prepared and a meeting held in the late afternoon. This meeting updates the status of jobs in progress and lists priorities for the night crew. This shift normally works from 1600 to 2400. Minor modifications are made throughout the day to the general work plan as new information is received and priorities change.

It should be mentioned that the scheduling process is made in a less than ideal decision making environment. Conventional configuration for a standard maintenance control is a large open office. It is the communications hub of the entire department. There is seldom any partitioning that would give some degree of privacy to those involved.

While the MCO/MCC attempts to make an optimal work schedule, phones may be ringing or parts arriving. Personnel may seek guidance from the decision makers, they may receive phone calls, or may be completely pulled away from the process by an urgent event or beckon from a superior.

It is under these somewhat adverse conditions that crucial scheduling decisions are often made. Unfortunately, the decision makers do not have the luxury of isolating themselves from the many disturbances or making the decision in an undisturbed environment. In fact, many intentionally attempt to make the schedule while still tuning into the conversations and happenings occurring around them, not wishing to lose contact with up to the minute events.

The morning scheduling process takes place in a less adverse environment, since it is largely formulated prior to the majority of worker's arrival for work. The afternoon and night-crew schedule formulations are not as fortunate. Conditions previously described are fully evident.

2. Priorities

Investigation determined that a number of rules and considerations are taken into account in deciding the priority of discrepancies. The material which follows is an attempt to structure into a basic classification scheme, the priorities most often used by the professional maintenance personnel interviewed. No attempt to give weighted values was made, although it is evident that such is the case for many of the rules when actually applied.

a. Flight Schedule Commitments

The squadron flight schedule is produced daily by the Operations Department and covers the next day's flights. It is a planning document that lists the mission number, pilots, takeoff and landing times, type of mission, and any special notes or configurations about the mission. Typically, there is a set of morning, afternoon, and night launches.

Meeting the flight schedule requirements with safely flyable aircraft assets is the number one priority for maintenance each day. All available resources will be expended to insure aircraft are preflighted and configured on time for each assigned flight.

The first consideration each morning is to assign aircraft to each event on that morning's flight schedule. These aircraft must be properly configured and capable of performing the assigned mission. Frequently, one or two aircraft are assigned as backups to replace any aircraft that go down prior to launch. Although this process is not a part of the discrepancy scheduling process, it has the effect of determining which aircraft are available to be worked on. Depending on the time of the next launch, additional aircraft may be set aside to meet later events. These aircraft are not usually available to be worked on either.

Should insufficient up aircraft be available for the next launch, the MCO/MCC is left with three possible alternatives. One of the up aircraft returning from the morning launch may be turned around and assigned the afternoon launch. A second alternative requires the remaining aircraft to be configured or repaired in time to be assigned the afternoon mission. In either case, a degree of uncertainty results, uncertainty the MCO/MCC prefers to not have to contend with. The third and least desirable alternative is to cancel the event.

b. Downing Discrepancies

As described earlier, downing discrepancies are those which prevent an aircraft from flying. A primary goal of maintenance is to minimize the number of down aircraft.

Of all the discrepancies on the squadron's aircraft, these receive a high priority.

A primary consideration when scheduling these discrepancies is the elapsed repair time for all the downing discrepancies against an aircraft. Besides the elapsed maintenance time for the repair itself, the time for any associated inspections or other actions are also included when determining the overall elapsed time requirements. For example, when an engine is changed, a special inspection requiring technical assistance must be performed. A test flight is also necessary prior to the aircraft being returned to an up status and released as safe for flight. The estimated time needed for these requirements is figured into the overall elapsed time to repair the discrepancy itself.

The aircraft whose total downing discrepancies require the least amount of estimated elapsed time for repair is normally worked on first. This is a primary rule used to decide on which discrepancy to work. Other considerations come into play when making this time estimate. These will be discussed later.

c. Up Discrepancies

There are many considerations when it comes to determining the priority of up discrepancies. Normally only gripes that are awaiting maintenance (i.e., not waiting for parts from supply) or are being trouble-shot to determine the cause of the problem are considered.

One method of differentiating is to weight PMC(M) discrepancies at a higher precedence than non-PMC(M). Again, as with downing discrepancies, one normally wishes to minimize PMC(M) gripes. Further prioritization often exists within the PMC(M) category itself. It is based upon the mission importance and frequency of use of a system. An example is giving a weighted advantage in scheduling to an IFF system discrepancy, which may be necessary for many missions, compared to a discrepancy on the FM radio, which is used rather infrequently. Both of these are PMC discrepancies. The system that is more important, in the judgment of the MCO/MCC, is given priority. This ranking process is used not only for determining the precedence of discrepancies against the same aircraft, but also in deciding between two aircraft with PMC(M) discrepancies.

Low priority up discrepancies are sometimes considered for aircraft which are projected to meet later launches in the day and for which there is minimal risk that working on the gripe could lead to the aircraft's downing. A low priority up discrepancy is one that is minor in nature and has a small impact on the aircraft's ability to perform its mission. Low priority up discrepancies have a low possibility of degrading an aircraft's status when trouble-shot or repaired, i.e., turning into down or PMC gripes. A discrepancy related to minor corrosion or a torn passenger seat are examples.

Another rule some units apply is to consider the age of the up discrepancy when determining priority. For instance, a lower priority gripe over thirty days old and still AWM, is given an increase in priority in a unit stressing no AWM gripes greater than thirty days old. Also, there may be a rule which states that an aircraft is placed on maintenance hold if it has greater than some number of AWM gripes, perhaps fifteen.

The two previous rules are established to insure minor up discrepancies don't become excessive. The previous rules are examples of the use of rules to determine priorities amongst up discrepancies. The weight each might receive may vary from squadron to squadron. Nonetheless, they are examples of additional rules which are frequently considered in scheduling prioritization.

d. Scheduled Maintenance/High Time Components

Maintenance planners must also consider scheduled maintenance and high time component changes in their overall work scheme.

(1) Scheduled Maintenance. Scheduled maintenance is maintenance which occurs at a set time. Examples are seven and fourteen day inspections or phased inspections (which are based on a certain flight hour interval). This maintenance is required and an aircraft is carried in a nonflyable status once it becomes necessary to perform this maintenance. To allow for some flexibility, these inspections may be waived for a day or for a short number of flight

hours, usually no greater than ten percent of the phase hour interval.

One consideration for phased or major flight interval inspections is the workload already assigned to the phase work center. Because of the manning of the work center and the nature of the work (usually performed in a sequenced manner) only one aircraft will be inducted into this work center at a time. This limitation is another constraint that must be considered when planning.

(2) High Time Components. Many major components on an aircraft are allowed a certain number of flight hours and then replaced or removed for inspection. This includes dynamic components, engines, generators, and transmissions.

To allow for flexibility in scheduling this maintenance, there frequently is a range of time during which they may be changed. For example, an engine may have a 600 hour limitation with a ten percent extension that allows it to be flown up to 660 flight hours after its installation.

High time component changes usually require considerable time and manpower for removal and replacement. They also frequently require a post maintenance functional test flight. They are often ordered at a low priority in advance of their change time. When the replacement component is received by supply, increased priority is frequently given to scheduling such maintenance, even though the required

elapsed repair hours may be greater than other rules which govern scheduling precedence.

e. Other Considerations

There are other factors which may influence the MCO/MCC's decision on what discrepancies to give preference. The following factors were considerations in the decision process with all the persons interviewed.

(1) SPINTAC. Aircraft that have not flown in sixty days are termed special interest aircraft (SPINTAC). The most common scenario that leads to an aircraft becoming SPINTAC commonly involves an aircraft going down for a major component and a rather long expected delivery date for the part. Such an aircraft frequently becomes a source for cannibalization parts for other aircraft. Because of cannibalization, it is not unusual for such an aircraft to end up with five to fifteen major parts on order against it.

Because of aircraft safety concerns, there has been high level interest in minimizing this category of aircraft. Increased supply attention is given for outstanding parts. There are pressures on all commanders to minimize SPINTAC aircraft. As a result, many commands take somewhat extreme actions to avoid allowing an aircraft to exceed this sixty day no fly period. This includes the cannibalization of major components not readily available in the supply system.

In many commands, aircraft which could go SPINTAC in anywhere from ten to fifteen days, high work priority is given to the down discrepancies. Even though giving such priority may be in contradiction to other rule considerations previously discussed, this priority is necessary if the aircraft is to be repaired and thus avoid SPINTAC status.

(2) Aircraft That Are "Flyers". In many squadrons one or two aircraft often seem to have the reputation for staying up and outflying other aircraft. They are the "flyers" of a squadron, and often receive priority in scheduling simply because of this reputation. No rational reason can be given for this. As in the case of SPINTAC aircraft, the priorities given such aircraft often take precedence over other rules used to determine priority in scheduling.

(3) Parts Received. Often, maintenance initially trouble-shoots a discrepancy and determines that a part is bad and must be replaced. At this point material control orders the part from supply. It may be in stock and delivered in an hour or so or it may not be in stock and have to be ordered from the supply system.

When the ordered part is received from supply, a weighted priority is often given to scheduling its installation. One reason for this priority is the fact that the initial troubleshooting has been completed and a

part determined to be the cause of the problem. The degree of uncertainty as to the cause is thus reduced. The MCO/MCC consequently has more definitive information with which to make his decision and a good likelihood that replacing the part will resolve the discrepancy.

A second consideration stems from the desire to minimize the amount of time a part is held before installation. This reduces the probability that the component may be lost, misplaced, or damaged if it sits around. The MCO has no desire to be a mini supply warehouse. For such reasons, when a part is received (even in the case of a low priority up discrepancy), it is frequently given assignment priority.

3. Constraints

Several constraints influence or restrict the scheduling of discrepancies. The MCO/MCC has a limited number of resources which he must efficiently use. Constraints may be classified as either fixed or variable. Fixed constraints are those that are basically unchanging and known by the planners. Hangar space and amount of Ground Support Equipment (GSE) are examples.

Variable constraints vary from day to day and hour by hour. They often involve a degree of uncertainty when considered. Personnel availability or technical representative assistance are examples.

In the material that follows, fixed constraints are discussed first, followed by variable constraints.

a. Hangar Space

Hangar space is a limited resource. In many units two or three aircraft are all that may be hangared at one time. A hangar may provide several benefits, such as protection from the elements, lighting for improved working conditions at night, or perhaps an overhead crane or high pressure air.

Closely related to this constraint are the work areas available on board ship when a squadron is deployed. This restriction is even more limiting because of the fewer available spaces to work on aircraft. This constraint must be taken into consideration when planning the overall maintenance schedule and priorities.

b. IMRL

Each squadron has an allocation of special tools for its type of aircraft based upon the Individual Material Readiness List (IMRL). This list is based on the number of assigned aircraft and the possible tactical missions with which the unit is tasked. Thus the number of special tools is limited. This restriction has to be taken into account when deciding the jobs to be assigned and the tools necessary to do the task.

c. PME/Test Equipment

Precision Measuring Equipment (PME) is the calibrated test equipment and tools a unit possesses. As with IMRL equipment, PME is limited. Because this gear must be

turned in periodically for calibration, the amount on hand may also vary.

d. GSE

GSE includes the tractors, electrical power and hydraulic units, workstands, etc. The amount of this equipment is limited and some is subject to mechanical failure and repair. This is another area that somewhat constrains the ability to assign discrepancies to be worked on. For example, two aircraft may each have down hydraulic system discrepancies that require use of a hydraulic test unit. If the squadron only has one hydraulic test unit available, one of the jobs that may originally have been given priority by other rules, is forced to be delayed until the other job is completed and the hydraulic unit freed.

e. Personnel

This is the first of the variable constraints to be described. How to employ all the personnel assets efficiently is a constant challenge for the MCO/MCC. Personnel available in the various work centers vary from day to day and hour by hour. Many factors affect this. When drawing up the work schedule the MCO/MCC must consider not only the number of personnel available, but also the technical capabilities and training of the personnel. These factors influence the estimate of the time to complete a task.

Valuable information on a work center's personnel situation requires good communications between the MCC and

the work center supervisor. Because the personnel factor is such a dynamically changing variable, it is constantly reassessed.

f. Type of Discrepancy

Some discrepancies restrict other discrepancies on the same aircraft from being worked on simultaneously. For instance, a discrepancy that will likely involve the breaking of a hydraulic line prohibits any electrical power from being applied to the aircraft because of the danger of fire. Therefore, for safety reasons, the MCC would not schedule an avionics discrepancy to be worked on at the same time as a hydraulic-related job.

Another consideration in this area involves assigning a job component that later has to be redone when another discrepancy is repaired. Two discrepancies, one of which called for the replacement of the electrical generator and the other requiring the transmission to be changed are an example. In the course of removing and replacing a transmission, the generator must be removed from the old transmission and installed on the new. Thus, it is better to first remove and replace the transmission and then replace the generator.

g. Local Constraints

Other factors occasionally influence work priorities. Noise abatement periods are sometimes issued. They preclude engine turnups or aircraft takeoffs during a

specified period. These periods are usually announced a few days in advance. They restrict certain types of maintenance that require power units or aircraft turnups.

For some units, especially those located in cold climates in the winter or extremely hot climates during the summer, restrictions will apply as to when or where aircraft may be worked on. These restrictions simply place additional constraints on the decision process.

h. Support

Some repairs require technical assistance from higher echelon maintenance activities. These jobs are thus dependent on this assistance being available. Close liaison and careful planning are necessary so that the necessary assistance is available to complete this job in a timely manner.

Technical representatives are frequently called in for assistance when a discrepancy is difficult to diagnose or fix. These personnel are very limited. Frequently one representative supports several squadrons. If this assistance is necessary, a delay of several hours is not unusual before a representative may be available for assistance. This delay must be planned for.

4. Knowledge Base

If one were to consolidate a knowledge base that the MCO/MCC draw from in the course of applying their scheduling techniques, it could be separated into three broad

classifications: prioritization rules, aircraft systems knowledge, and parts availability.

a. General Prioritization Heuristics

The various prioritization heuristics, discussed above, are a critical element of the MCO/MCC's knowledge base. They are used for determining what jobs to work on and the priority to assign them. These rules are joined in different combinations and are given different weights. The research conducted in this study has only touched on the very basic and elementary rules used for decision making. They are broad enough to substantiate the complexity of the scheduling process in this domain and provide an understanding of the method priorities are determined.

b. Aircraft Systems Knowledge

An aircraft system is composed of the major functioning components that constitute the aircraft. This knowledge consists of information about the major parts that are combined to form a system, as well as technical knowledge of the functioning of the system itself.

For example, a typical UHF radio system is composed of several major components which include such items as the radio transmitter and receiver unit, the frequency control box, the antenna, and the coaxial cable that connects the components.

Some major components have subsystems associated with them that are themselves systems. The aircraft

engine system consists of the major components of the engine, as well as such subsystems as the start system and the fuel system.

The MCO and MCC draw on their knowledge of the aircraft's systems in the prioritizing process. Two concepts related to systems are frequently applied in formulating the maintenance schedule. First is the history of the system itself with respect to a particular aircraft model. Has the system had a frequent failure rate? Does a particular component of the system have an unusually high history of failure? Secondly, the history of the system associated with the individual aircraft is considered. Has this system failed recently on this aircraft? What action was taken to repair the first instance? How long ago was the repair completed? Is it a repeat discrepancy or similar to the previous discrepancy?

Incorporating this type of consideration into the decision strategy may allow the MCO/MCC to make his decision from a more informed point of view. This type of information is part of the expert's knowledge base. Such information on systems is not instantly available to an expert. It is acquired over the course of several months or several years experience with a particular aircraft model. When initially making decisions on an aircraft model with which the expert has not gained such systems knowledge, uncertainty increases for the decision process. Although specific information of this nature is available from the specialists who repair

the system, the MCO/MCC seldom has the luxury to call on these personnel for every system he may be uncertain about. A decision is made with the available knowledge.

A second systems-related concept is the time to complete a repair. Over time, MCO/MCCs build up a general knowledge of the elapsed time requirements to complete a repair. This includes time necessary for troubleshooting, removal and replacement of the faulty component, and any necessary inspections required. Expected repair times play a key role in deciding on which discrepancies to give priorities. This is discussed in greater detail in Chapter V.

c. Parts Availability

There are two sources of parts available to replace a faulty component. The normal, and most frequently used is the supply system. The MCO/MCC is concerned whether a particular part is available at the local supply level or whether a requisition has to be sent off station to procure the part. This is factored into their judgment in deciding which jobs to give priority. Discrepancies for which the expected parts are readily available are given a higher priority.

Even though a certain part is not usually available from supply, the MCO/MCC may still consider working on the discrepancy, knowing that he may cannibalize the likely part from another down aircraft. Considerations

include the amount of time and effort necessary to remove such a part and how fragile the part is to possible breakage or damage during the process of removal. Some components are never cannibalized because of this.

d. Personnel Capabilities

Another bank of knowledge often used by maintenance controllers concerns the personnel resource they have to work with. Enough work, but not too much, must be assigned. Not only a knowledge of the number of personnel available in a given work center, but their general level of technical competence, are used in the decision making process.

The degree of training and knowledge a work center's personnel have, directly affects other decision factors. For example, if a work center has many new personnel not trained or familiar with an aircraft's systems, it can be expected that additional time is required to fix a given discrepancy. This significantly influences the elapsed time of repair consideration and is used in assigning the quantity and priority of discrepancies.

5. Conclusions/Comments

a. Complexity of the Decision Process

The discussion in this section points out the complexity of the discrepancy scheduling process. The decision maker must attempt to balance and tradeoff any number of dynamic factors in making a master work schedule.

Analysis indicates that the process can be broken down into a taxonomy of basic heuristics, with knowledge of a particular aircraft model applied.

A fundamental question asked of the experts during the course of the interviews, dealt with whether the same intrinsic decision process would apply if they were switched to another type of aircraft. All strongly agreed that it would. The factors that would change are the constraints and knowledge of the new aircraft's systems. Changes to the rules and their basic decision making methodology would only be slightly modified.

b. Difficulties Encountered

The interviews with the maintenance professionals were conducted to uncover the basic concepts and strategies they use in this domain. It must be recognized that the concepts and factors expressed here represent only a fraction of those used by the decision makers in solving the problem of what to schedule for work. Nevertheless, the formalization attempted here should be a good starting point for future work in this domain. It also serves to point out that the decision process is relatively structured.

Waterman points out that it is seldom effective to ask the expert to directly express the rules and methods used for solving the problems in their domain [Ref. 10: p. 153]:

"Experts," it appears, have a tendency to state their conclusions and the reasoning behind them in general

terms that are too broad for effective machine analysis. It is advantageous to have the machine work at a more basic level, dealing with clearly defined pieces of basic information that it can build into more complex judgments. In contrast, the expert seldom operates at a basic level. He makes complex judgments rapidly, without laboriously reexamining and restating each step in his reasoning process. The pieces of basic knowledge are assumed and are combined so quickly that it is difficult for him to describe the process. When he examines a problem, he cannot easily articulate each step and may even be unaware of the individual steps taken to reach a solution. He may ascribe to intuition or label a hunch that which is the result of a very complex reasoning process based upon a large amount of remembered data and experience. In subsequently explaining his conclusion or hunch he will repeat only the major steps, often leaving out most of the smaller ones, which may have seemed obvious to him at the time. Knowing what to consider basic and relevant and not requiring further reevaluation is what makes a person an "expert."

This quote concisely describes the difficulties encountered in the course of attemptint to discover the factors that make up the discrepancy scheduling process for the domain. In fact, at the conclusion of the interview, each interviewee was asked to read this passage and all agreed it expressed the exact difficulties they had wrestled with in preparing for the interview.

The next chapter examines the NALCOMIS hardware and software assets that are proposed for installation in every aviation squadron. Many of the requirements of the system are designed to be of aid to the MCO/MCC.

IV. NALCOMIS

The mission of operational maintenance and material units is to maximize aircraft mission readiness by maintaining high material condition standards [Ref. 21:p. 1]. In the mid-1970s it was determined that the existing manual information system was inadequate to support the effective management of Naval Aviation maintenance and material, especially with the manpower and fiscal restrictions then in effect.

The manual system was slow, onerous, and labor intensive. Management requirements in the maintenance and material support areas were becoming increasingly complex as sophisticated aircraft weapon systems entered the inventory and as the operational tempo of units increased. NALCOMIS was proposed as a means of providing a modern, responsive computer-based Management Information System (MIS) for this domain. The scope of the system is limited to support of the organizational maintenance activity (OMA), aircraft intermediate maintenance department (AIMD), and supply support center (SSC) [Ref. 22:p. 2-1].

This chapter focuses on NALCOMIS at the organizational maintenance level, its history, components, and the current status of the prototype. Possible modifications and changes in configuration to the NALCOMIS system are addressed. No

attempt is made to analyze the current NALCOMIS hardware and software prototype configuration, however. Such analysis is beyond the scope of this research and, furthermore, the latest prototype software for the OMA has not yet been delivered.

A. HISTORY

Development Milestone I for NALCOMIS was approved in February, 1977. This milestone required the project to use SNAP hardware, authorized development of a system design, and selected MCAS Cherry Point, NC as the prototype site. It also specified that the operational prototype system must be approved prior to installation of hardware at any other site [Ref. 21:Encl. (1) p. 1].

In January, 1979, approval of Milestone II permitted the full scale development and testing of the prototype system. Because of delays in the procurement of the SNAP hardware, development of the software was begun on a Perkin-Elmer minicomputer. It was not until June, 1982 that the SNAP contract was awarded to Honeywell Information Systems, Inc. July, 1983, saw the delivery of a Honeywell minicomputer to the prototype site. The converted Perkin-Elmer software, termed NALCOMIS Standard Environment, proved to be inefficient when run on the Honeywell hardware. Unacceptable terminal response times were the most serious problem.

A competitive contract was issued in late 1983 for redesign of the application software. The new prototype software is to be delivered prior to the end of this year and will undergo several months of testing and evaluation.

In the interim, there has been a proposal to permit deployment of the AIMD software module, which has been tested and certified. This deployment concept has been approved and allows the procurement and deployment of the hardware necessary for this implementation. It is uncertain how much of the requested funding will be approved for the project in the FY-86 budget.

Upon successful completion of Milestone II, Milestone III will seek deployment of the hardware required for full implementation at the OMA, AIMD, and SSC for all approved NALCOMIS sites [Ref. 21:p. 2].

B. OBJECTIVES

The NALCOMIS Mission Element Needs Statement identified three major management deficiencies at the OMA, AIMD, and SSC. They were: a lack of real-time management information, a difficult data collection process, and inadequate and inaccurate upline information [Ref. 21: Encl. (1): p. 1].

The current information system does not accommodate the efficient processing of the mass of available raw data in the timely and coherent fashion needed for real-time

decision making. A second shortcoming is the inability of the present system to support responses to individual queries in an accurate and timely manner [Ref. 23:p. I-6].

The present data collection process is largely manual. It is labor intensive and there are significant error rates in the data reported. Frequent updating and revalidation are necessary. Finally, most of the information provided by the data is out of date.

Upper level commands suffer from the incomplete, erroneous, and untimely data of the present reporting system. This seriously affects higher echelon's ability to manage logistical demands, budget justification, personnel staffing, etc.

Objectives to correct each of these major shortcomings are established for NALCOMIS. The following is a list of the minimum specific objectives for NALCOMIS [Ref. 23: p. I-9]:

- Provide timely and accurate information to maintenance and material managers to improve their effectiveness.
- Improve the number of FMC aircraft.
- Reduce the NMCS and NMCM rates.
- Reduce the supply response time when maintenance requisitions parts.
- Respond more quickly when maintenance demands requisition status for parts on order.
- Achieve a reduction in beyond the capability of maintenance (BCM) actions at the AIMD for components which may be repaired locally.

- Improve the visibility of critical rotatable pool items.
- Reduce the maintenance and supply personnel man-hours required for data collection, entry, and validation.
- Improve the quality and timeliness of data used for upline reporting.
- Reduce awaiting parts inventory levels at the SSC.
- Reduce the administrative burden of maintenance personnel in meeting 3M system requirements.
- Provide local control of information.

C. HARDWARE

The material presented in the following two sections relates to OMA requirements and is largely condensed from Reference 3. The hardware is a ruggedized version of off the shelf commercial equipment. It incorporates the architecture of the Honeywell DPS 6 system.

This DPS 6 system consists of 16 and 32 bit processors. The OMA version is designated the DPS 6/54 model and has one Mbyte of memory, expandable to two Mbytes. This model uses an asynchronous bidirectional bus architecture and can support up to forty communications lines. Mass storage units, printers, communications controllers, etc., may be attached.

Cycle time is 300 nanoseconds. Direct memory access is used for all data transfers. There is a tie-breaking network which prevents lock-up of the bus. The central processor timing unit is set at five microseconds (Figure 4.1.a).

HONEYWELL DPS 6/54 MINICOMPUTER

16 Bit Processor

1 Mbyte Memory (Expandable to 2 Mbyte)

300 Nanosecond Cycle Time

Asynchronous Bidirectional Bus

a. Computer Hardware Specifications

52 Mbyte Winchester Disk Drives

Tape Drive

2 Diskette Drives

Video Display Terminals

Printer

b. Peripherals

GCOS6 MOD 400 Operating System

Honeywell IDS-II Data Base Management System

ANSI-COBOL-74 Compiler

Application Software

c. Software

Figure 4.1 NALCOMIS Hardware and Software

Memory save is provided by a battery backup system. Primary storage is provided by six Winchester disk drives, each of which has 52 Mbytes of storage capacity. Each squadron also has a tape drive for historical storage of data. Two diskette drives, video display terminals, and a printer are also included in the system configuration. This material is condensed in Figure 4.1.b.

D. SOFTWARE

This section briefly describes the software used by NALCOMIS. This can be divided into two categories, Honeywell developed software and the NALCOMIS application software now in prototype development.

1. Honeywell Software

To minimize project risk, off the shelf software was provided by Honeywell. Honeywell furnished the operating system, data base management system, transaction processing system, and compilers (Figure 4.1.c).

The operating system is the GCOS6 MOD 400. It is a real-time disk-oriented system which allows interactive dialogue for multiple users. Both real-time activities and batch processing may be run concurrently.

The Honeywell Integrated Data Store (IDS-II) data base management system is used for the system. This system provides real-time and multiuser capability and serves to control communications between data in the mass storage units and the user. Data integrity, independence, and

security are provided by the system. A query language is furnished in the form of GEN 5 software [Ref. 22:p. 4-3]. The data manipulation language is COBOL-based.

Transaction processing allows for the scheduling, loading, and execution of real-time programs. The Honeywell Interactive Transaction Processing System satisfies this requirement. It supports data base and file sharing among multiple users.

An ANSI-COBOL-74 compiler is used since all programs must be written in COBOL. This is in keeping with government requirements. Maintenance and software updates are furnished by Honeywell Information Systems under contract with the government.

2. Application Software

The initial application software developed was termed the NALCOMIS Standard Interface. As previously mentioned, it proved unsatisfactory, and a contract was let for a new version of the application software. This new software is given the name "native mode" and is designed specifically for the DPS 6 system. The OMA version is to be delivered for prototyping in late 1985.

The OMA software may be broken into the following eight functional subsystems:

- Flight Activity Subsystem
- Maintenance Activity Subsystem
- Configuration Management Subsystem

- Maintenance Personnel Management Subsystem
- Asset Management Subsystem
- Supply Support Center Subsystem
- Local/Upline Reporting Subsystem
- System Support Subsystem

Additional information on these various subsystems is available in Reference 2.

E. EXPANDABILITY

One of the key advantages of the Honeywell DPS 6 system is its capability for modification. The main memory of the DPS 6/54, used at the OMA, may be expanded rather easily and economically to two Mbytes. Additional memory expansion up to 16 Mbytes may be possible. Winchester disks may also be added for increased mass storage. Although not part of the present contract, supplemental compilers are available. These include higher level language compilers for FORTRAN, BASIC, and C.

V. A KNOWLEDGE BASE FOR THE MAINTENANCE SCHEDULING DOMAIN

Chapter II defined knowledge base as that portion of an expert system that contains the facts and heuristics of the problem domain. In many cases the knowledge base, i.e., the domain specific facts and rules, may be the key difference between two different expert systems. For two systems developed using the same expert system building tool, only the knowledge bases differ. This chapter presents a recommendation for a basic knowledge base necessary for an expert system for scheduling discrepancies in aviation maintenance and implemented with NALCOMIS. Chapter II listed rules, aircraft systems knowledge, and parts availability information as key items in a knowledge base for this domain. These items also underlie the discussion of the knowledge base in this chapter. The discussion begins with general comments on planning and scheduling, the generic category of expert systems under which this problem falls.

A. PLANNING/SCHEDULING DOMAINS

The development of planning or scheduling expert systems has been primarily theoretical and research lab oriented. Wilensky, Sacerdoti, and Stefik have written books on the subject of planning, but these works are not specifically related to the problem domain being studied [Refs. 24,25, 26]. There are a few articles on the planning process in

general which have gone so far as to build models. Again, these articles had little relation to the problem posed [Refs. 27,28].

The only article that could be found that deals exclusively with planning and scheduling describes a research system termed ISIS [Ref. 29]. It is a knowledge-based decision support system for job shop scheduling. In some ways this problem domain is similar to aircraft maintenance scheduling. SRL, a frame-based language, was used to build the ISIS system [Ref. 29:p. 30]. Although this is the only example of the scheduling task that could be found, it does lend support that it is possible to develop expert systems in a complex scheduling domain.

One other potential source of information was discovered as this thesis was in the finishing stages. A brochure for the First International Expert Systems Conference, to be held 1-3 October 1985 in London, listed one of the presentations as "An Expert Fuzzy Planner for Scheduling Aircraft Repair Work." Squadron Leader T.J. Grant of the Ministry of Defence was the speaker. An attempt to obtain reference materials on this research proved unsuccessful.

B. KNOWLEDGE FROM THE PRESENT DATA BASE

Hayes-Roth points out that it has become common for knowledge systems to access and retrieve information from on-line data bases [Ref. 30:p. 15]. Other works have pointed out the practical benefits to be gained, strategies

for extracting data, and the research challenges presented in coupling an expert system to a data base [Refs. 31,32]. An expert system for this domain requires data from the NALCOMIS data base be used extensively.

The requirement to use NALCOMIS obviously adds complexity to designing a system for the maintenance domain. Nevertheless, it is far too inefficient to consider any other way of acquiring the facts needed for the knowledge base. The redundancy of data is unnecessary and undesirable.

The following paragraphs look at the data in the NALCOMIS data base that are related to a maintenance scheduling knowledge base. It should be remembered that a knowledge base consists of facts and heuristics. The NALCOMIS data base contains only some of the facts but none of the rules required for the knowledge base.

1. Aircraft Facts

The data base has hundreds of facts that might be used by an expert system. The majority of this data provides facts related to aircraft. The following is a list of potential aircraft facts from the data base:

- Aircraft Bureau numbers/side numbers
- Discrepancy Facts
 - * Aircraft Type/Model
 - * Category (NMCM, NMCS, PMCM, etc.)
 - * Description of Malfunction
 - * System Affected (Work Unit Code)

- * Date of Discrepancy
- * Work Center Assigned
- Last Fly Date
- Last Scheduled Maintenance Date/Flight Time
- Aircraft Configuration

2. Other Facts

In addition to aircraft-related facts, the NALCOMIS data base contains other data that an expert system might access. Some of these are listed below:

- GSE
 - * Type of Equipment
 - * Amount of Equipment
- IMRL/PME
 - * Type of Tools/Test Equipment
 - * Amount Available

C. KNOWLEDGE BASE

The knowledge base for this domain consists of additional facts not contained in the NALCOMIS data base plus the heuristics used in determining priorities.

1. Facts

Many facts not available from the data base are necessary to express the expert knowledge in this domain. For example, a file of the parts received for outstanding discrepancies is needed. This information is used when considering the priority given to jobs for which parts have been received.

A listing of the precedence given to PMC discrepancies and their weighting should be resident in the knowledge base. The discrepancy on the IFF taking priority over one a seldom used FM radio was an example of this given in Chapter III. An additional fact not in the present data base is the amount of hangar space available. This information is necessary so the system considers this constraint and doesn't exceed the limitation.

There is a need to quantify the other constraints covered in Chapter III in the knowledge base. One consideration is to specify any special tools or equipment that are required when a discrepancy is put into work. It might be best to include this type of information with that initially captured when the discrepancy information is originally entered into the data base. The expert system considers this information to ensure any required special equipment is available prior to a discrepancy being assigned. For example, once all the special tools of a particular type are assigned to jobs to be put in work, another discrepancy needing such a tool would not be assigned or considered for assignment until the expected completion time of one of the previously assigned jobs.

One of the key factors in prioritizing jobs is considering the expected total elapsed repair time for the discrepancy. These figures are presently nonexistent for the different aircraft or systems. This information is

determined primarily from experience gained over a period of time from similar problems associated with a specific system and model of aircraft.

When a discrepancy is first received, the MCO/MCC attempts to form an intuitive estimate of what the cause of the problem is and the time necessary to repair it. The normal process considers each of the possible failure components, an estimate of the probability of each component being the cause, and the estimated elapsed repair time for each possibility. As previously mentioned, the time for any special inspections or check flights required for the repair are also factored into the computation of the overall expected elapsed time for repair.

Portions of this problem lend themselves to possible solution using algorithmic programming methods. Nevertheless, such methods need to know the estimated repair time for each component, as well as the probability of the component's being the cause of the system problem. This information is not currently published or available.

One possible source for this information is the Naval Aviation Logistics Data Analysis (NALDA) System. NALDA is an operational automated information system. Its primary mission is to provide information to support the Naval Aviation logistics community. It is the central data base repository for aviation logistical data.

NALDA has a data base for each type aircraft by model. To understand this organization better, it is worthwhile to briefly explain aircraft type and model. Examples of aircraft type are A-4 or CH-46 aircraft. The model is a one letter character which further delineates a type of aircraft into a subseries of production aircraft. Different model aircraft of the same type may be quite different, e.g., they may have different engines, different radios or navigation equipment, or different weapon systems. A CH-46D has a number of significant differences from a CH-46E.

Inquiry was made into the possibility and difficulty of extracting weighted probabilities for the rate of failure for each component in a system. It was also asked if it was possible to determine an average elapsed repair time for each major component or the repairable subassemblies of a given system. [Ref. 33]

NALDA's data base is ideally organized for our purposes, since we want to query the system for the historical background on a part as used in a particular type and model aircraft, and not its history of use on all types of aircraft. Response to the questions in the previous paragraph indicated that the required estimated elapsed repair time and weighted failure rates of components within a system can be determined from the NALDA data base.

2. Heuristics

The other part of the knowledge base is the heuristics or rules section. The term "rules" is used as a general term throughout this section, as there are several methods of representing knowledge, as previously discussed. It should be recalled from Chapter II that "frames" are another representation. From the ISIS example presented earlier and writings on planning, frames appear to be a preferred representation scheme for this category of knowledge-based systems.

The rules of a knowledge base use the facts of the knowledge base as the basis for making decisions. It is the inference engine of the system that decides how to apply the rules and in what order.

Many of the general rules that the MCO/MCC use to determine their priorities were generally stated in Chapter III. These rules were captured from the initial interviews with the maintenance experts, and it should be stressed, only represent a surface level of knowledge of this domain.

It is beyond both the scope of this research, as well as the implementation stage of development that the research represents, to attempt to establish even a small number of the specific rules that apply to this problem domain. Rather, general rules may be used as a starting point from which to explore the more complex interrelationships that exist and from which further research may proceed.

Figure 5.1 lists general rules that would form the bedrock of the knowledge base. These rules are taken from the heuristics stated in the interviews and presented in Chapter III. Rules that represent knowledge of the problem domain but are not recommended for inclusion in the knowledge base (explained in Sections C. and D. that follow) are not listed.

D. KNOWLEDGE FROM USER QUERY

Some facts are too dynamic to attempt to keep updated in the knowledge base. The system should get this information by querying the user just prior to running the system. Two good examples of this kind of information are the aircraft assigned to fly (and therefore not available to be worked on) or the number of aircraft currently in the hangar. The system needs to consider this information. User query seems the most accurate and efficient way to provide it.

There may be other factual information the system needs from time to time to clarify a point or to continue processing. User query is a method of providing this information so that an accurate final output is provided. Nevertheless, consideration must be given to minimizing this method for critical information items. Nonessential queries not only drastically increase the system utilization time but also demand valuable time from the decision maker.

IF:	A/C is NMCM, and OERT < six hours	
THEN:	Downing DISCRP = High	
IF:	A/C is NMCM, and Last fly date is > 45 days (i.e., nearing SPINTAC)	
THEN:	Downing DISCRP = High	
IF:	A/C is NMCM, and A/C is a "flyer", and OERT is < 18 hours	
THEN:	Downing DISCRP = High	
IF:	A/C is NMCM, and DISCR is high time component change	
THEN:	Downing DISCRP = Medium High	
IF:	A/C is NMCM, and OERT > six hours	
THEN:	Downing DISCRP = medium high	
IF:	A/C is PMCM, and PMC code = High (/medium/low)	
THEN:	DISCRP = Medium High (/medium/medium)	
IF:	A/C is FMC, and status changes from AWP to AWM	
THEN:	DISCRP = Medium	
<u>ABREVIATIONS</u>		
IF:	A/C is FMC, and DISCR > 30 days	A/C = aircraft
THEN:	DISCRP = Medium	AWM = awaiting maint.
		DISCR = discrepancy
		DISCRP = discrep. priority
		OERT = overall elapsed
IF:	A/C is FMC, and DISCR = AWM	repair time
THEN:	DISCRP = Low	

Figure 5.1 Discrepancy Scheduling Rules

E. KNOWLEDGE NOT CONSIDERED

There are two factors that are best not incorporated into the knowledge base, although they are considered in making a final decision on what work to assign. These factors are personnel and common sense maintenance knowledge.

1. Personnel

Regarding personnel, there are two primary considerations that are used in deciding what and how much work to assign. The number of personnel available to work varies considerably throughout a day. Because this is such a dynamic factor it should not be factored into the expert system for consideration. There are any number of reasons for personnel fluctuation. Some personnel may be sick and not present. Others may have a medical or other type of appointment which requires their absence; still others may be assigned to a working party. Some are on leave or temporary additional duty assignment.

Another limitation is associated with the technical proficiency of the personnel. This may be a cumulative estimate that the MCO/MCC considers as a general guideline in assigning the amount of work to a work center as well as the ability of the personnel to meet expected elapsed repair times. It may also take the form of direct communications feedback from a work center supervisor stating that the mechanic with the real technical expertise is not present and recommending a job be delayed.

2. Common Sense Maintenance Knowledge

This subject was briefly touched on in Chapter III under the topic "Type of Discrepancy." It refers to the very broad spectrum of general knowledge about aircraft maintenance. It is this kind of knowledge which states that a hydraulics repair cannot be performed at the same time as an avionics discrepancy. The fact that a radio is an electrical component and requires electrical power to check the system is common sense knowledge. We cannot efficiently hope to capture and represent all of this type of knowledge. The order in which the components of a system are assembled is another example of common sense knowledge. Because of the vast amount of this knowledge, there is no effective way of incorporating it into the knowledge base.

Supporting the recommendation that the above two knowledge areas be excluded initially from the knowledge base is the fact that most developed expert systems have not attempted to capture all the knowledge of a particular domain. The reason for this is because the state of the technology is not sufficiently advanced to do this. Systems that have performed well have taken rather restricted tasks and applied only the key knowledge of the domain. The items stressed in Chapter II are very applicable to these issues.

Furthermore, a MCO/MCC that is given a priority work schedule produced by an expert system can quickly factor

in any modifications that are necessary because of considerations in these areas. For instance, given a list that included both the hydraulic and avionic discrepancies mentioned earlier, with both at the same priority level, the MCO could quickly spot the conflict and ensure only one of the discrepancies was assigned. An initial system in this domain should not necessarily try to encompass 100 percent of the domain knowledge.

VI. FEASIBILITY ANALYSIS

This chapter presents an analysis of the thesis research. It addresses the three major issues posed during the course of the investigation: Are expert systems applicable for the aircraft maintenance scheduling problem domain? Can NALCOMIS provide the technological support for an expert system? Is development of an expert system warranted?

A. EXPERT SYSTEM APPLICABILITY TO MAINTENANCE SCHEDULING

This section examines the suitability of an expert system for the aircraft discrepancy planning process. The benefits of using a knowledge-based system are considered as well as the drawbacks related to developing such a system.

1. Suitability of the Problem Domain

An expert system can be developed and would prove worthwhile for aiding the maintenance control scheduling problem. There are several sources which provide items to consider when evaluating whether an expert systems approach is appropriate for a given problem [Ref. 10:pp. 127-134; Ref. 14:p. 160; Ref. 2:p. 198]. The following points covered in these sources are directly relevant to the maintenance control problem domain.

Do experts exist? The research conducted here clearly indicates there are people in the field that are generally acknowledged as having a degree of knowledge

significantly higher than others. They are noted not only for their decision making ability and knowledge in the planning domain, but also in the other areas of responsibility associated with maintenance control. The interviews indicate that these experts are able to articulate the methods they use and generally agree on the process and heuristics used in making a decision.

Does the problem require common sense? AI programming techniques are unable to represent common sense knowledge very well [Ref. 10:p. 129]. By restricting the size and complexity of the problem domain, as recommended in Chapter V, the present task does not include common sense. Cognitive, not physical, skills are necessary.

Is the task too simple or too difficult for an expert system to solve? "Too simple" a task is one classified as requiring the expert but a few minutes; "too difficult" a problem needs from a few days to a month to solve [Ref. 10: p. 128]. A task that requires from thirty minutes to several hours to be resolved is acceptable for today's developmental capabilities. This problem falls within this guideline, taking from thirty minutes to an hour.

Other factors point to this problem domain as acceptable for expert system solution. The potential improvement in operational readiness is a substantial payoff. This type of expertise is also required in all aviation units, not just a few. Waterman cites expert systems as

being justified in situations where expertise is lost through personnel changes.

Retirement, job transfer, and military duty reassignment often cause disruption and even havoc because of the vital expertise that experienced personnel take with them when they leave. The institutional memory aspect of an expert system can minimize or even eliminate this problem. [Ref. 10:p. 131]

After analysis, it is evident that this problem is heuristic in nature. Rules of thumb are used extensively to reach a solution. These rules are identifiable and therefore facilitate building a knowledge base. Although parts of the problem domain may lend themselves to solutions by conventional programming techniques, the problem as a whole does not. It is too dynamic and complex. These factors all favor an expert systems approach as being a viable solution to the problem at hand.

The knowledge in this problem is symbolic rather than numerical. It is subjective, judgmental, and changing. These knowledge characteristics all point to artificial intelligence (AI) techniques rather than algorithmic solutions.

2. Benefits from Developing an Expert System

Development of an expert system for this problem domain would provide several benefits for the users. Such a decision aid augments the human capability and productivity in maintenance control. It allows the expertise from many human experts to be combined into a shared knowledge base. This rare and costly expertise, acquired after years of

experience, may then be widely disseminated. In developing such a system, the knowledge is formalized and clarified.

There are several composite squadrons in the Navy and Marine Corps. These units have several type of aircraft assigned to them. Such squadrons could significantly benefit from a system which is able to apply expertise about each type aircraft to the scheduling problem.

Development expenses would be minimized because of the wide distribution of the system. Changes would not be extensive for systems supporting another type of aircraft. Many of the heuristics would be the same; many of the facts would already be resident in the NALCOMIS data base. Today's aircraft are frequently kept fifteen to twenty years. Only gradual changes would be necessary to an expert system as modifications were made to the aircraft. A knowledge-based system also produces more consistent and reproducible results than does a human expert.

Finally, such a decision support aid would allow the MCO/MCC to concentrate more time on other pressing problems. These potential benefits are very significant when one considers that the maintenance field is already limited by personnel and material constraints. It is unlikely manpower in a squadron will be increased or that more parts will be available. Development of an expert system offers one of the few methods for potentially achieving significant gains in aircraft operational readiness under the existing

constraints. Even a small gain of as little as 2 percent, when applied across the entire naval aircraft inventory, translates into an additional one hundred operationally ready aircraft each day.

3. Drawbacks

Development of an expert system for this problem is not without some drawbacks. Expert systems are expensive to develop. Although the operating hardware is already available, some expense may be incurred for possible modifications in the areas of memory expansion, mass storage capability, or the addition of another compiler. Development of an operational system could take as much as four to five man years of effort before system performance is reliable [Ref. 34:p. 39].

Because of the lack of AI compilers for NALCOMIS hardware, the developed system would require translation into a high level language for which a compiler is available. While this provides wide transportability of software, it does restrict the ability of local modification of the programs. It should also be mentioned that some risk is obviously involved in developing leading edge software. The discussion on planning and scheduling in Chapter V makes this evident.

B. NALCOMIS CAPABILITY TO SUPPORT AN EXPERT SYSTEM

The literature in the knowledge engineering field provides little information on hardware and system requirements

for running an expert system. Discussions do state that development of such systems is done using AI workstations which provide a symbolic language development environment. It is also pointed out that programs developed in LISP or PROLOG are frequently translated into a higher level language for use on available user systems. Nevertheless, when it comes to providing operational requirements for expert systems nothing specific could be found.

A phone interview with Dr. Nelson Marquina, a knowledge engineer with Honeywell Systems and Research Center, proved most enlightening. The basic problem area and the NALCOMIS hardware were described to him. Estimates of from 1000 to 2000 rules for the knowledge base were assumed. It was also assumed that from 100 to 200 discrepancies were outstanding and that no other demands would be simultaneously made on the system.

Dr. Marquina was asked to estimate hardware memory requirements and the time to process the data and produce an output. Stating that the figures were only rough estimates, he suggested that one-half Mbyte of memory was required and from five to ten minutes were needed to run the program. To be on the safe side, given the impreciseness of the assumptions, he recommended one Mbyte of memory. Dr. Marquina also stated that these figures were based on the assumption that the program was written in an efficient higher level language, such as C, and that the rules were considered nontrivial.

While these figures must be looked on as inexact, they nonetheless provide some guidelines to the requirements for such a system.

From a hardware perspective NALCOMIS meets the basic memory estimate of one Mbyte. There is little doubt that additional mass storage capability, in the form of additional Winchester disks, would be necessary. NALCOMIS at the OMA level is capable of being expanded in this area.

NALCOMIS presently uses COBOL as its higher level programming language. Both a literature search and numerous interviews with people working in the knowledge engineering field failed to find one application where COBOL had been used as a translation language for an expert system originally developed using a symbolic language. If an expert system were to be developed for NALCOMIS a high level language compiler other than COBOL would have to be used to run the expert system. There is a capability to add a different compiler to the DPS 6 system.

Another aspect to consider for improving the efficiency of an expert system is the use of on-call procedures which are more effective at compiling some aspects of a problem than symbolic programs. Algorithmic programming techniques for determining the expected elapsed work hours is an example of a situation where this could prove beneficial.

A final question arises. Can NALCOMIS afford to lockout its basic functions as an MIS for two or three ten minute

periods required to run the expert system in the course of a day? There is no other dedicated machine to download NALCOMIS data on. The initial run in the morning, before most work centers are using the system, does not present a problem. It is also reasonable to believe that the NALCOMIS system would not be adversely effected by the other expert system runs. Although NALCOMIS provides real time access to the data base, this information is seldom so critical to any maintenance function that it must be instantly available. Should an exceptional reason arise that necessitates instant access, the expert system run could be aborted and run later.

In summary, NALCOMIS is likely to support an expert system with only slight modifications to the present system's architecture. Minimal degradation of the functions provided by the MIS may result.

C. IS DEVELOPMENT OF AN EXPERT SYSTEM JUSTIFIED?

An expert system does provide a service for which a need really exists. This need is for more effective decisions to be made when scheduling maintenance. One might ask are all maintenance control work centers equally capable? The overwhelming opinion of professionals is no, they are not. A key factor determining the quality of this work center is the ability of the MCO and MCC to make effective decisions. This ability varies. There are some however, who are clearly considered "experts." The decisions the

MCO and MCC make in planning the work schedule will have a significant impact on resulting aircraft operational readiness. Any decision support tool for this area can provide valuable assistance.

Chapter III covered several negative aspects of the decision environment in maintenance control. These factors would have little effect on an expert system. Use of the expert system would also nullify much of the lost expertise caused by the turnover of key personnel inherent with the military profession.

Can the human decision maker in maintenance control, assimilate, and synthesize all the information available? Studies by cognitive scientists have shown that human memory consists of clusters of symbols called "chunks." Chunks are hierarchically organized collections of symbols. Research has concluded that a human can only maintain and process from four to seven chunks in short-term memory at one time [Ref. 2:p. 24]. Vast amounts of facts and rules must be taken into consideration when scheduling. There is good reason to believe there is more information available than can be comprehended and compiled by the average maintenance controller in making a decision. An expert system can use and process all the available information and therefore make a more knowledgeable decision.

The NALCOMIS project was approved over eight years ago. At that time it proposed a state of the art MIS. Today's

prototype gives every indication that it does an excellent job of meeting its objectives. Nevertheless, since NALCOMIS's inception new technologies have entered the work place; first in the form of decision support systems and followed by today's expert systems. As late as 1980 the majority of expert systems were still in the research labs [Ref. 2:p. 1]. Today many systems, and the tools for building these systems, are being rapidly developed. It is very likely that NALCOMIS has the technological capability to support this state of the art technology. Moreover, much of the knowledge required for the knowledge base is already resident in the present data base. It seems only logical to use both the hardware and data assets to their fullest. Further development of potential uses for NALCOMIS should not stagnate while prototyping continues. Research on new and innovative technological applications should simultaneously be pursued.

There are two other indirect benefits of developing an expert system for this problem domain. First, further investigation of the problem will undoubtedly provide valuable insight and greater understanding of the scheduling process itself. Weiss and Kulikowski state that from a scientific point of view, the most important reason for building an expert system is the formalization and clarification of knowledge that results from having the human expert make his reasoning explicit [Ref. 1:p. 7].

A second important benefit is the introduction of AI technology in the military at a very visible but noncrucial level. There currently exists considerable resistance on the part of military users, and in society as a whole, to accept any technology with the AI label. MYCIN has been proven to be as capable as experts in the medical profession at diagnosing infectious blood diseases. However, it has never been widely accepted or used by the medical profession.

This same resistance exists for military applications of AI. Users will not freely accept the introduction of such systems into critical life and death or tactical applications. Pilots will be unwilling to turn over to a computer the flying of the aircraft in a crisis situation. Naval officers will likewise be hesitant to trust the defense of the ship, including weapons response, to a computer. Current levels of resistance not only delay research funding in these areas, they often lead to scrapping of projects altogether. It is contended that for more technically advanced AI systems to gain acceptance, practical non-critical AI systems must first be introduced to the users. As users gain familiarity and confidence in the more general and small applications, they will be more willing to accept and pursue technologically advanced projects. An expert system applied to the maintenance scheduling problem seems to be just the right type of project. It deals with a nontrivial and

complex decision process and would expose numerous personnel to AI technology.

VII. CONCLUSIONS AND RECOMMENDATIONS

The scheduling decisions made in maintenance control are crucial to the aircraft operational readiness of a squadron. These decisions are currently made using the techniques developed and used twenty years ago. While these techniques are functionally sound, they do not provide the capability to fully use all available information in determining a decision. The decision environment is simply too demanding and complex to do so.

The development of an expert system for prioritizing aircraft discrepancies to be worked on is both feasible from a technological standpoint and desirable because of the improved decision support it would provide. The problem domain is suitable for expert system development. There are "experts" in the field. The task is sufficiently complex and difficult for expert system application. The planning problem is generally heuristic in nature and requires symbolic rather than numerical solution.

Development of such a system would improve the efficiency and effectiveness of the scheduling decisions made in maintenance control. Improved operational readiness is a direct result. It allows available information to be used more fully in the decision process. The expertise of several experts is combined in a shared knowledge base. Such a system

also minimizes the negative effects caused by the loss of expertise resulting from the frequent personnel reassignments inherent with military service. It provides aircraft historical knowledge to the MCO/MCC transferred to a new aircraft type. An expert system could be widely distributed throughout Naval aviation with only aircraft specific knowledge changing.

While no definitive answer can be given as to the capability of NALCOMIS to support such a system, there are several favorable indicators that suggest that it could. The expandability of NALCOMIS hardware and peripherals can provide both the necessary memory and mass storage requirements an expert system requires. Although the present COBOL-based software is not acceptable for expert system implementation, there are other suitable compilers available and compatible with the DPS6/54 system.

There are two negative factors which need to be considered. The costs and time to develop an expert system for this problem domain are not trivial. The potential improvement in operational readiness, however, more than offsets these factors. The implementation of an expert system with NALCOMIS would likely require the lockout of normal system functions for short periods two or three times per day. The information provided by NALCOMIS is not of such a critical nature that these few delays are not acceptable.

Development of an expert system for aircraft discrepancy scheduling has many potential benefits. It provides state of the art decision support for the user. It allows a greater use of the data available from NALCOMIS. The potential for improved aircraft readiness is substantial. It permits the introduction and wide visibility of AI technology at a practical level, setting the groundwork for more critical AI applications in the future. Further research in this area is warranted.

During the course of this research, several areas were examined and related recommendations are in order. The potential application of a knowledge-based system to the maintenance control scheduling domain should be more thoroughly investigated by knowledge engineering professionals.

Other areas that offer benefits for effective maintenance decision making should be explored. Operations research tools are one possible source. System component failure rates and elapsed repair times, as discussed in Chapter V, should be extracted from NALDA or other aviation data bases and made available to maintenance decision makers.

Other possible uses of the data and hardware assets provided by NALCOMIS should be explored. Many new software productivity and decision making aids have been developed since the original inception and design of the system several years ago. Although the OMA portion of this program is still in the prototype development stage, other possible applications should be considered.

Writings on the acquisition and coding of knowledge and on the implementation requirements of a knowledge-based system need to be published. There is a considerable amount of literature on expert systems scattered in books, technical reports, conference proceedings, etc. Unfortunately for one considering the potential implementation of an expert system, they are either written at a theoretical or at a general informative level. Practical writings on the acquisition, formalization, and coding of knowledge for expert systems are necessary. Technical information on the hardware and software requirements for implementing expert systems is virtually nonexistent. Information in this area is needed.

Based on the previous discussion, it is submitted that development of an expert system for scheduling discrepancies is both feasible and appropriate. It should be emphasized that such a system would serve as a decision support tool and not as a replacement for the MCO/MCC's decision making for this domain. The improved management effectiveness and potential for improved aircraft operational readiness that an expert system offers are well worth the costs.

LIST OF REFERENCES

1. Weiss, S.M. and Kulikowski, C.A., A Practical Guide To Designing Expert Systems, Rowman and Littlefield, 1984.
2. King, D. and Harman, P., Expert Systems, John Wiley & Sons, Inc., 1985.
3. Hayes-Roth, F., "Knowledge Based Expert Systems," Computer, Vol. 17, No. 10, Oct 1984, pp. 263-273.
4. Newell, A. and Simon, H.A., Human Problem Solving, Prentice-Hall, 1972.
5. van Melle, W., Shortliffe, E.H., and Buchanan, B.G., "EMYCIN: A Knowledge Engineer's Tool for Constructing Rule-Based Expert Systems," in B. Buchanan and E. Shortliffe (eds.), Rule-Based Expert Systems, pp. 302-308, Addison-Wesley, 1984.
6. Bachant, J. and McDermott, J., "R1 Revisited: Four Years in the Trenches," The AI Magazine, Vol. 5, No. 3, Fall, 1984.
7. Duda, R.O. and Reboh, R., "AI and Decision Making: The PROSPECTOR Experience," in W. Retiman (ed.), Artificial Intelligence Applications for Business, Ablex, 1984.
8. Barr, A. and Feigenbaum, E.A., The Handbook of Artificial Intelligence, Vols. I and II, William Kaufman, 1982.
9. F. Hayes-Roth, Donald A. Waterman, and Douglas B. Lenat, "An Overview of Expert Systems," in F. Hayes-Roth, D.A. Waterman and D.B. Lenat (eds.), Building Expert Systems, Addison-Wesley, 1983.
10. Waterman, D.A., A Guide to Expert Systems, Addison-Wesley, 1985.
11. Stefik, M., Aikins, J., Balzer, R., Benoit, J., Birnbaum, L., Hayes-Roth, F., and Sacerdoti, E., "Basic Concepts for Building Expert Systems," in F. Hayes-Roth, D.A. Waterman and D.B. Lenat (eds.), Building Expert Systems, Addison-Wesley, 1983.
12. Kinnucan, P., "Software Tools Speed Expert System Development," High Technology, March 1985, pp. 16-20.

13. Buchanan, B.G. and Shortliffe, E.H., Rule-Based Expert Systems, Addison-Wesley, 1984.
14. Buchanan, B.G., Barstow, D., Bechtal, R., Bennett, J., Clancey, W., Kulikowski, C., Mithcell, T., Waterman, D.A., "Constructing an Expert System," in F. Hayes-Roth, D.A. Waterman and D.B. Lenat (eds.), Building Expert Systems, Addison-Wesley, 1983.
15. Gaschnig, J., Klahr, P., Pople, H., Shortliffe, E., and Terry, A., "Evaluation of Expert Systems: Issues and Case Studies," in F. Hayes-Roth, D.A. Waterman, and D.B. Lenat (eds.), Building Expert Systems, Addison-Wesley, 1983.
16. Kinnucan, P., "Computers That Think Like Experts," High Technology, Jan 1985, pp. 30-42.
17. van Emden, M.H., "Chess Endgame Advice: A Case Study In Computer Utilization of Knowledge," in D. Michie (ed.), Introductory Readings in EXPERT SYSTEMS, Gordon and Breach Science Publishers, Inc., 1982.
18. Chandrasekaran, B., "Expert Systems: Matching Techniques to Tasks," in W. Reitman, (ed.), Artificial Intelligence Applications for Business, Ablex, 1984.
19. Hayes-Roth, F., "The Industrialization of Knowledge Engineering," in W. Reitman, (ed.), Artificial Intelligence Applications for Business, Ablex, 1984.
20. Chief of Naval Operations, OPNAVINST 4790.2C, The Naval Aviation Maintenance Program (NAMP), Vol. II, 11 June 1984.
21. Naval Air Systems Command, NALCOMIS System Decision Paper, 9 April 1984.
22. Fleet Material Support Office, NAVAIRSYSCOM Document No. 41L001 FD-A-01 Revision 3, NALCOMIS Module 1. Independent Functional Process "A". Detailed Functional Description, 1 December 1984.
23. Naval Air Systems Command, NALCOMIS Preliminary Prototype MAG 14 Evaluation Report (Draft), January 1985.
24. Wilensky, R., Planning and Understanding, Addison-Wesley 1983.
25. Sacerdoti, E., A Structure for Plans and Behavior, Elsevier, 1977.

26. Stefik, M.J., Planning With Constraints, Ph.D. Dissertation, Stanford University, Palo Alto, California, 1980.
27. Duersch, R.R. and Wheeler, D.B., "An Interactive Scheduling Model for Assembly-Line Manufacturing," International Journal of Modelling & Simulation, Vol. 1, No. 3, 1981, pp. 241-245.
28. Doumeingts, G., Pun, L., Mondain, M., Breuil, D., "Decision Making System for Production Control Planning and Scheduling," International Journal of Production Research, Vol. 16, No. 4, July-Aug. 1965, pp. 244-254.
29. Fox, M.S. and Smith, S.F., "ISIS--A Knowledge-Based System for Factory Scheduling," Expert Systems, Vol. 1, No. 1, July 1984, pp. 25-49.
30. Hayes-Roth, F., "The Knowledge-Based Expert System: A Tutorial," Computer, Vol. 17, No. 9, Sep 1984, pp. 11-28.
31. Jarke, M. and Vassiliou, Y., "Coupling Expert Systems with Database Management Systems," in W. Reitman, (ed.), Artificial Intelligence Applications for Business, Ablex, 1984.
32. Mittal, S., Chandrasekaran, B. and Sticklen, J., "Patrec: A Knowledge-Directed Database for a Diagnostic Expert System," Computer, Vol. 16, No. 2, Sep 1984, pp. 51-58.
33. Beall, S., Interview with the Director NALDA, NAF Pax River, April 1985.
34. Winston, P.H. and Prendergast, K.A., The AI Business, The Commercial Uses of Artificial Intelligence, MIT Press, 1984.

INITIAL DISTRIBUTION LIST

	No. Copies
1. Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6145	2
2. Library, Code 0142 Naval Postgraduate School Monterey, California 93943-5100	2
3. Professor Norman H. Lyons, Code 54Lb Department of Administrative Sciences Naval Postgraduate School Monterey, California 93943-5100	1
4. Professor Daniel R. Dolk, Code 54Dk Department of Administrative Sciences Naval Postgraduate School Monterey, California 93943-5100	1
5. BGEN Ross S. Plasterer Assistant Wing Commander 3d Marine Aircraft Wing MCAS El Toro Santa Ana, California 92709	1
6. CAPT W.T. Jordan PMA-270 Naval Air Systems Command Headquarters Washington, D.C. 20361	1
7. Dr. G. DeJong Naval Research Laboratory, Code 7510 4555 Overlook Avenue, S.W. Washington, D.C. 20375	1
8. LtCol. Don Frost Aircraft Maintenance Officer 3d Marine Aircraft Wing MCAS El Toro Santa Ana, California 92709	1
9. MAJ John Keiffer NALCOMIS Project Officer MAG-14 2d Marine Aircraft Wing MCAS Cherry Point, North Carolina 21576	1

10. MAJ Marty McCaffrey
Headquarters (LMC-2)
U.S. Marine Corps
Washington, D.C. 20380-0001

4

Thesis
M1634
c.1

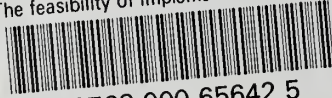
McCaffrey

214449

The feasibility of
implementing an expert
system for aircraft
maintenance discrepancy
scheduling with the
Naval Aviation Logis-
tics Command Manage-
ment Information Sys-
tem (NALCOMIS).

thesM1634

The feasibility of implementing an exper



3 2768 000 65642 5

DUDLEY KNOX LIBRARY